

Netherlands  
organization for  
applied scientific  
research

TNO-report

**FEL**

**AD-A263 347**



**TNO Physics and Electronics  
Laboratory**

①

**P.O. Box TD 92-3274**

2509 JG  
Oude Waalsdorperweg 63  
The Hague, The Netherlands

Fax +31 70 328 09 61  
Phone +31 70 326 42 21

title

report no.

**FEL-91-B293**

copy no.

13

**Secure Open Systems,  
An Investigation**

Nothing from this issue may be reproduced  
and/or published by print, photoprint,  
microfilm or any other means without  
previous written consent from TNO.  
Submitting the report for inspection to  
parties directly interested is permitted.

In case this report was drafted under  
instruction, the rights and obligations  
of contracting parties are subject to either  
the 'Standard Conditions for Research  
Instructions given to TNO' or the relevant  
agreement concluded between the contracting  
parties on account of the research object  
involved.

TNO

author(s):

**P.L. Overbeek**

NTSC  
APR 26 1993  
D

date

**December 1991**

**TDCK RAPPORTCENTRALE**

Fréderikkazerne, gebouw 140  
v/d Burchlaan 31 MPC 16A  
TEL. : 070-3166394/6395  
FAX. : (31) 070-3166202  
Postbus 90701  
2509 LS Den Haag TDCK

classification

title : unclassified  
abstract : unclassified  
report text : unclassified  
appendix A : unclassified

no. of copies : 21  
no. of pages : 108 (incl appendix, excl RDP and distr.list)  
appendices : 1

STATEMENT  
for public release  
Unlimited

All information which is classified according to  
Dutch regulations shall be treated by the recipient in  
the same way as classified information of  
corresponding value in his own country. No part of  
this information will be disclosed to any party.

93 4 26

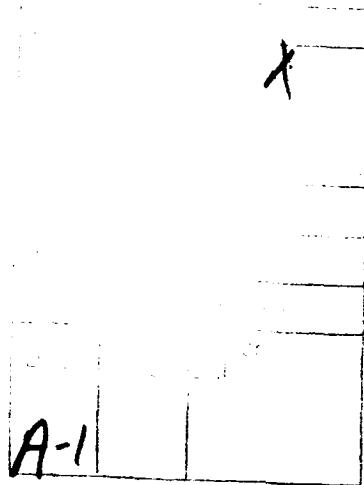


**93-08833**



report no. : FEL-91-B293  
title : Secure Open Systems, An Investigation  
author(s) : P.L. Overbeek  
institute : TNO Physics and Electronics Laboratory  
date : December 1991  
NDRO no. :  
no. in pow '91 : 709.2  
Research supervised by : D.W. Fikkert, H.A.M Luijff  
Research carried out by : P.L. Overbeek

=====



#### ABSTRACT (unclassified)

This report outlines the achievements of current standardisation efforts in the area of secure open systems. Security in open systems is a special problem since all elements in an open system (hardware, networks, operating systems, databases and other applications) must be able to offer the required security in co-ordination with each other.

First, a new view on requirements for security is presented. Security requirements are studied from different angles: security requirements that are specific to open systems, security requirements that stem from organisational considerations, security requirements that reflect the value of information and services for an organisation, security requirements that stem from social structures and, finally, security requirements that address the security of the system itself.

Next, the results of our investigation of current initiatives in the area of technical security in open systems are presented. Among others, the standardisation initiatives of CCITT, DoD/NCSC, CEC, ECMA, IEEE, ISO and NATO are studied. These initiatives are placed within the context of a simple model of systems in a network.

#### The main conclusions are:

- None of the initiatives addresses all basic requirements for secure open systems.

- None of the initiatives gives a solid basis for co-ordination of security among all elements of an open system.
- None of the initiatives regard the security functionality that is needed to map organisational structures and responsibilities.
- The needs for security that stem from society are hardly addressed.
- The basic security functionality of the system offered by the initiatives is rather divergent and sometimes conflicting. Emphasis is put on prevention. Other security measures are neglected to a large extent, and, if addressed at all, they lack structure.
- The initiatives show a lack of integration among application security, operating system security and network security.
- It is concluded that an architecture is needed that crosses the borders of the elements of an open system. This architecture must define the security functionality in and between the elements of the open system. This requires properly defined interfaces between the elements of an open system.

---

rapport no : **FEL-91-B293**  
titel : **Veilige 'Open Systemen', een onderzoek**  
  
auteur(s) : **Ir. P.L. Overbeek**  
instituut : **Fysisch en Elektronisch Laboratorium TNO**  
  
datum : **december 1991**  
hdo-opdr no :  
no. in iwp '91 : **709.2**

Onderzoek uitgevoerd o / v : **D.W. Fikkert, ir. H.A.M. Luijff**  
Onderzoek uitgevoerd door : **Ir. P.L. Overbeek**

---

#### **SAMENVATTING (ongerubriceerd)**

Dit rapport beschrijft de huidige situatie op het gebied van de standaardisatie-initiatieven voor beveiliging in 'open systemen'. Beveiliging in open systemen is een bijzonder probleem omdat alle elementen van een open systeem (applicaties, besturingssystemen en netwerken) samen in staat moeten zijn om in onderlinge samenwerking de noodzakelijke beveiliging te bieden.

Allereerst wordt een nieuwe visie op beveiligingseisen gegeven. Beveiligingseisen worden bestudeerd uit verschillende invalshoeken: de beveilingseisen die specifiek zijn voor open systemen, beveilingseisen die voortkomen uit de noodzaak organisatorische verhoudingen te weerspiegelen in een systeem, beveilingseisen die gebaseerd zijn op de noodzaak om de waarde van informatie en -diensten voor de organisatie te beschermen, beveilingseisen die voortkomen uit maatschappelijke verhoudingen en, ten slotte, beveilingseisen die ingaan op de bescherming van het systeem zelf.

**Vervolgens worden de resultaten van het onderzoek naar de huidige initiatieven op het gebied van standaardisatie van technische beveiliging in open systemen gepresenteerd. Dit betreft onder andere standaardisatie-initiatieven van CCITT, DoD/NCSC, CEC, ECMA, IEEE, ISO en NATO. Deze initiatieven worden in de context geplaatst van een eenvoudig model voor systemen in een netwerk.**

**Enkele belangrijke conclusies zijn:**

- Geen van de initiatieven voldoet aan al de elementaire beveiligingseisen voor beveiliging in open systemen.
- Geen van de initiatieven geeft voldoende mogelijkheden voor een gecoördineerde aanpak van beveiliging tussen alle elementen in een open systeem.
- Geen van de initiatieven biedt goede voorzieningen om organisatorische structuren en verhoudingen te representeren in het systeem.
- De initiatieven gaan nauwelijks in op de beveiligingseisen die voortkomen uit maatschappelijke verhoudingen.
- De initiatieven hebben zeer verschillende en soms conflicterende benaderingen in de voorziening van de meest elementaire beveiliging. De nadruk ligt op preventie. Andere mogelijkheden voor beveiliging krijgen geen, of geen gestructureerde aandacht.
- De initiatieven bieden te weinig mogelijkheden voor integratie van beveiliging in applicaties, het besturingssysteem en het netwerk.
- Er is een architectuur nodig die de grenzen van de elementen in een open systeem overschrijdt en de beveiligingsfunctionaliteit in en tussen de elementen beschrijft. Dit vereist tevens goede interfaces tussen de elementen van een open systeem.

## CONTENTS

	<b>ABSTRACT</b>	2
	<b>SAMENVATTING</b>	4
1	<b>INTRODUCTION</b>	8
1.1	Security	8
1.2	Information system	9
1.3	Definitions and terms	9
1.4	Structure of this report	9
2	<b>SECURITY REQUIREMENTS FOR (OPEN) SYSTEMS</b>	10
2.1	Organisation and Security requirements	10
2.2	Requirements for security of services and information from the perspective of owners and users	11
2.3	Requirements for security imposed by society	13
2.4	Requirements for the security of the system	14
2.5	Summary of security requirements	14
3	<b>A MODEL OF SYSTEMS IN A NETWORK</b>	17
4	<b>INITIATIVES IN THE AREA OF SECURE OPEN SYSTEMS</b>	20
4.1	Applications	21
4.2	Operating systems	21
4.3	Networks	22
4.4	General	22
4.5	Other initiatives	22
4.6	Map of initiatives: studied aspects	24
4.7	Guide to the readers	25

5	INITIATIVES THAT ADDRESS SECURITY IN APPLICATIONS	27
5.1	ECMA Framework for Secure Open Systems	27
5.2	CCITT Distributed Applications in Open Systems (DAF)	38
5.3	Trusted Database Management System Interpretation of the TCSEC (TDI)	41
5.4	Application-dependent security: FTAM, EDI, MHS and The Directory	47
6	INITIATIVES THAT ADDRESS SECURITY IN OPERATING SYSTEMS	49
6.1	Trusted Computer System Evaluation Criteria (TCSEC)	49
6.2	POSIX Security Interface	58
7	INITIATIVES THAT ADDRESS SECURITY IN NETWORKS	67
7.1	OSI Security Architecture	67
7.2	NATO OSI Security Architecture (NOSA)	73
7.3	Trusted Network Interpretation of the TCSEC (TNI)	74
7.4	MIT Athena Project: Kerberos	81
8	INITIATIVES THAT ADDRESS SECURITY IN SYSTEMS AS A WHOLE	84
8.1	Information Technology Security Evaluation Criteria (ITSEC)	84
9	DISCUSSION AND CONCLUSIONS	90
9.1	Fulfilment of security requirements	90
9.2	How do the different initiatives fit together?	92
9.3	Mutually beneficial approaches as a starting point for secure open systems	94
10	ACRONYMS	96
11	REFERENCES	98
APPENDIX A: INDEX		

## 1 INTRODUCTION

Currently, there is a drive towards *open systems*. There is no agreed definition of what an *open* system should be. Regrettably, it shares this with many terms in high-fashion information technology. During the mid-eighties, *open* was equivalent to Open Systems Interconnection (OSI) [12]. Later on the discussion focussed on UNIX<sup>1</sup> as an *open* operating system. Just a few years ago, the development of standard interfaces between applications and (proprietary or 'closed') operating systems gave us a new view on *open*-ness. A hardware architecture is said to be *open* when its interfaces are available to all interested parties. Recently, the so-called fourth-generation languages were introduced. The suppliers claim that these languages enable the development of *open* software, which means software that is independent of, say, a specific database management system.

Thus, *open* appears to be a moving target. In general, the following relate to *open*:

- Properly defined interfaces, services and protocols.
- Availability of these definitions to third parties.

Following the literature, the term *open* is used in this report in combination with elements of a system like hardware, networks, operating systems, databases and other applications.

## 1.1 Security

The elements of an open system must not only be able to coexist with one another but should also be able to benefit from one another and offer a concerted "value-added" effort. This also implies the co-ordination of security between the elements of an open system and consistency of security within an element.

It must be assumed, and this is not specific to *open* systems, that the information-technology infrastructure is shared with unreliable and unpredictable participants (computers, networks, users and software). Currently, information flow is not restricted to one specific computer system or network and not even to any specific application. Information security must secure the information at all times and ubiquitously. Therefore all information flow must also be secure. In order to achieve this in an open-systems environment, all elements of the open system must seamlessly fit together: standardisation is therefore essential.

## 1.2 Information system

So far, the term *information system* (or just *system*, for short) has been used in a intuitive way. There are many definitions of the term *information system*. In this report the following definition of an information system is used: a set of one or more services, the associated computers, peripherals, storage media, terminals, means for information transfer, etcetera, that forms one *autonomous* whole capable of performing information processing, storage and/or transfer. This definition is derived from [12]. Note that more systems together may form a 'super'-system and that within one system subsystems may exist that function autonomously with regard to specific information.

## 1.3 Definitions and terms

Different definitions of the same terms are not uncommon in the world of information technology. This report just marks the differences. Whenever possible, all terms will be used according to their definition in the referenced context. In other cases the ISO definitions as stated in [9] are used.

## 1.4 Structure of this report

This report outlines the achievements of the standardisation efforts in the area of secure open systems. First, requirements for secure open systems are discussed. Next the most important initiatives are discussed. Emphasis is put on standardisation efforts for technical security at an architectural level by organisations with a major impact on international developments like ISO, the European Computer Manufacturers Association (ECMA), the Comité Consultatif International Télégraphique et Téléphonique (CCITT), the USA National Computer Security Center (NCSC), the USA Department of Defense (DoD) and the North Atlantic Treaty Organisation (NATO). To obtain a map of their activities, they are placed within the context of a simplifying model of systems in a network.

This report concentrates on the security functionality itself. Non-technical aspects with regard to this functionality lie outside the scope of this report.

## 2

## SECURITY REQUIREMENTS FOR (OPEN) SYSTEMS

In this section, the security requirements for (open) information systems are studied from different angles. First, security requirements that come from organisational considerations are discussed. Next, the security requirements for information and services in the system are addressed, seen from the perspective of the users. These perspectives are chosen in such a way that all relationships that influence the security requirements for a system are covered.

The security requirements for *open* information systems are not different from those of other information systems. The difference between open and other systems is in the implementation of the requirements. An open system may consist of many elements (hardware, networks, operating systems, databases and other applications). These elements are called *open elements*. Each open element must be able to offer security in concert with the other open elements. It is not known beforehand which other open elements will be present in the system(s) where an open element will eventually be used.

## 2.1 Organisation and Security requirements

### 2.1.1 Security must fit the organisational structure

Each employee performs one or more roles (or: functions) in the organisation. He has been assigned tasks and responsibilities by the management of that organisation. Managers must be able to control the tasks for which they are responsible. For this, they need management information about these tasks (for example, information concerning the status, the progress and budgets of a project). Management tasks are special since they influence the tasks and responsibility assignments directly. Examples of management tasks are: definition of new tasks, authorisation of tasks to employees, auditing of the continuation of the assigned tasks, modification of task definitions, reassignment of tasks, termination of tasks and withdrawal of responsibilities.

The security offered by the information systems used by an organisation must fit the security requirements of that organisation. Therefore, it must be possible to express the employees' responsibilities and tasks in the real organisation in these information systems. At least the roles of users within the system should not be conflicting with the roles in the organisation, e.g. the

'controlled' in the organisation should not have ultimate 'control' using the system (see also [3, 59]).

Without such a mapping of real-life tasks and responsibilities to the information systems, these systems will only be of limited use as a management tool and additional technical and procedural measures must be taken to audit the employees' activities in the system.

#### 2.1.2 Representation of roles in a system

The 'real-life' roles are to be translated into roles in the system.

- An employee is represented in the system by one or more processes acting on his behalf (sometimes these processes are somewhat misleadingly called 'the user').
- Tasks are performed using services offered by the system. To enable an employee to perform his tasks, he is allowed (or constrained) to use certain services offered by the system. Through these services he is able to access information.
- Responsibilities are translated into rights and duties in the system.
- Task management implies that, using a management service, the rights and duties to perform services are (re-)assigned to employees. These management services also assure that the aggregation of certain responsibilities within one person, which is undesirable in real life, is also prohibited by the system. This is often referred to as *separation of duties*.

The progress of the tasks may be checked by means of a management service giving aggregated task information.

#### 2.2 Requirements for security of services and information from the perspective of owners and users

Information and services are valuable and therefore need be secured. Their value is largely determined by the well-known properties (taken from 66)):

- 1 *Confidentiality* is the exclusivity and exclusive use of information and services.
- 2 *Integrity* is the correctness and completeness of the information and services as well as the information and services being up-to-date, c.q. being the most recent version.
- 3 *Availability* is the ability to have access to services and information within a certain time frame.

Information security is not a goal by itself. Information security must reflect the needs of the organisation. Therefore, the relative value of information and services to the organisation must be taken into account when security measures are selected.

### 2.2.1 Security Measures

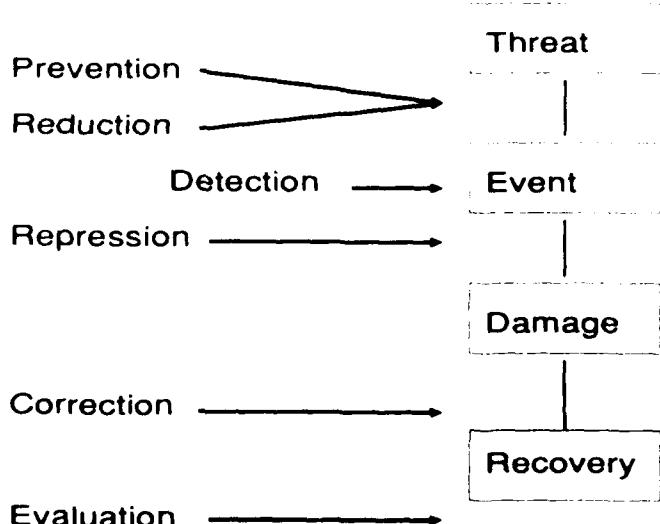
Security measures are most commonly subdivided into organisational, technical, physical and procedural security measures.

Another approach is to look at a chain of security measures that address a possible breach of security. In this report a breach of security is called a *security event*.

There are four different stages in the occurrence of a security event, see figure 1. These stages together are called the *event cycle*. At the top of the event cycle is the threat that security may be breached. A security event may cause damage as loss of information or services that requires recovery. These stages all need attention when security measures are planned.

Specific security measures can be applied to each stage. We have identified six types of security measures, see figure 1. First, the occurrence of a security event can be excluded or prevented by *preventive* security measures. At the same time, the possible loss resulting from an anticipated

## Event Cycle



Stages in the (possible) occurrence of a security event and the specific security measures applicable to the different stages

Figure 1: Event cycle

event can be minimised by *reductive* security measures. When an event occurs, it must be discovered. This is done by *detective* security measures. *Repressive* security measures stop the continuation or recurrence of an event, thus reducing losses. Next, the information and services are restored as well as possible by *corrective* security measures.

In case of a major event, it is worthwhile to *evaluate* the event: what went wrong, how did it happen and which corrective actions should be taken. Beside this evaluation on a per-event basis, it is helpful to have an organisation-wide view on all security events. One way to achieve this is by means of a reporting procedure for security events. These reports assist in the evaluation of the effectiveness of the current security measures and act as input for the updating of the security plan, viewed as the whole of security measures.

### 2.3 Requirements for security imposed by society

All organisations are part of social structures and are influenced by external relations. Examples are relations with: shareholders, management of the holding company, external accountants or auditors, the judiciary, (for banks:) the national central bank, external users (clients, suppliers) and relations with people about which information is stored in the information systems (the registrees). These are all external parties that are involved in the way an organisation handles its information systems. The resulting, sometimes conflicting, demands will impact the security requirements for the information systems of such an organisation.

Examples are:

- External clients may demand *anonymity*, while the service-providing organisation needs accountability (specifically *billability*).
- In supplier/purchaser relations, *proof of transactions* may be required.
- Many countries have legislation which regulates the *privacy* of information, implying technical security, amongst others. Privacy is an aspect both for those that are registered as well as those using the system (to what extent is it permissible to analyse the activities of employees?). Note that privacy encompasses both confidentiality and integrity of personal data.
- In many cases information about the information system itself and the security of the information system is required by external parties. This may concern the *proof of proper functioning* of the system (does it offer the correct figures?) and proof of certain activities (will the audit file be usable to provide *legal proof* of an action?).

## 2.4 Requirements for the security of the system

One of the key issues to security as seen from the system is that the system does not (have to) trust its users *a priori*. Furthermore, the system must be able to maintain its own security. Therefore, the system has to *control the use of services* (which may be trusted by the system or not). In doing so, access to information as well as other services is controlled. To make sure that a user remains within his sphere of responsibilities and tasks (mapped to rights and duties in the system), the system must know on whose behalf a service, or, in more general terms, a security-relevant action, will be performed. This requires user *identification* and *authentication*. Every security-relevant action is mapped against the assigned *rights and duties* of the user. The scope of what a security-relevant action is depends on the *granularity* in the system, i.e. the definition of what can/must be managed as a whole, seen from the standpoint of security. The granularity is determined by the units of information that can be managed individually (data field, record, file, database, file structure, etc.) as well as the active entities that can be distinctly managed in the system (e.g. processes, applications, services, 'pipes').

In the safe situation the system can be trusted by its users and by the owner of the information, which in most cases is the organisation. By controlling all security-relevant actions, the system is able to maintain a safe situation. In doing so, it is also able to secure itself, which is essential for continuation of the secure situation.

## 2.5 Summary of security requirements

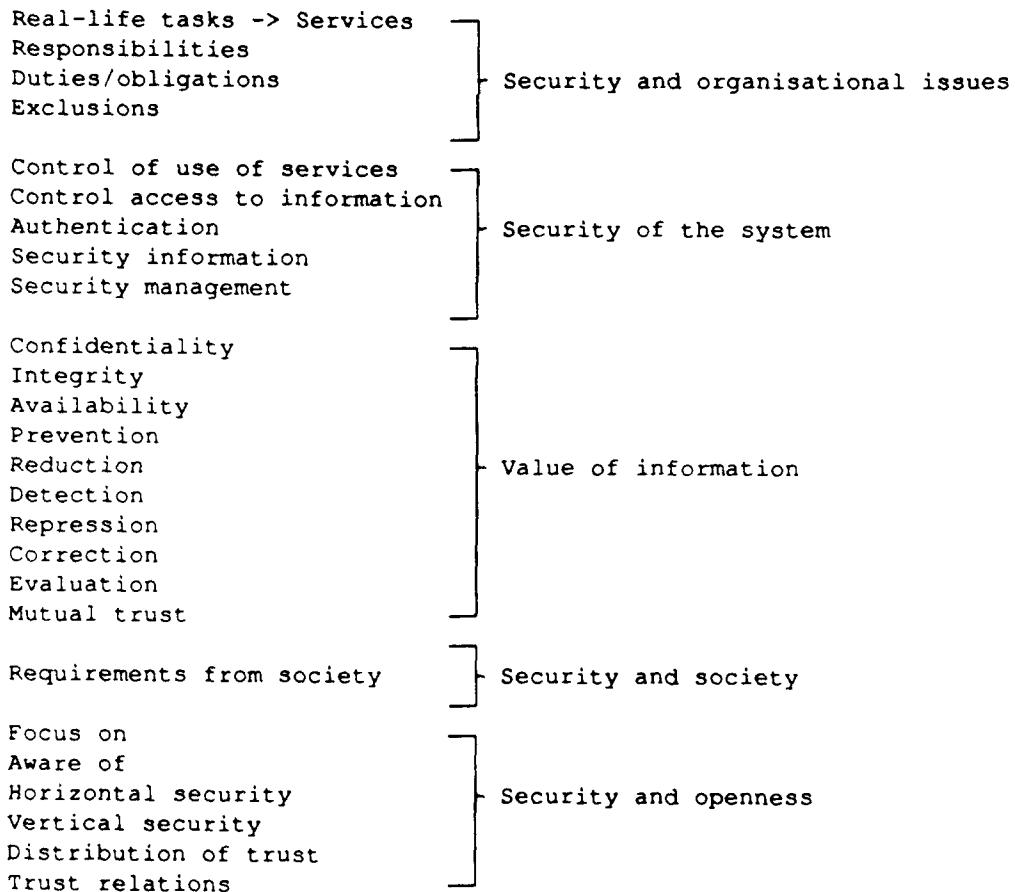
In the next sections our investigation of current standardisation initiatives in the area of secure open systems is presented. Each of the identified initiatives is studied using the following checklist to see which security requirements are addressed:

- Security requirements that stem from organisational considerations:  
It must be possible to represent organisational structures and relations as well as the users' real-life tasks and responsibilities in the system. Considerations are: the representation of an employee in the system; the mapping of tasks to services by which information can be accessed and/or handled; the mapping of responsibilities to rights and duties in the system; the mapping of management tasks to services in the system.

- Security requirements for the security of the system:  
The system must be able to maintain its own security. Therefore, the system must be able to control every security-relevant action (involving services or information). The granularity in the system is a yardstick for what should be a security-relevant action. Security information is needed about the information in question, the assignment of rights and duties concerning the identified user, the service and the information as well as security information about other ongoing activities in the system.
- Security requirements that regard the value of information:
  - Information and services are valuable and therefore need be secured. Their value is determined by the properties confidentiality, integrity and availability.
  - The security measures must reflect the value of the information. In the chain of security measures that address a possible security event we recognise measures that aim at prevention of an event, reduction of the consequential losses of an anticipated event, detection, repression and correction of a security event as well as measures that contribute to the evaluation of security events.
  - It is required that an organisation can trust the system in the way it handles the organisation's valuable information.
- Requirements for security imposed by society:  
All organisations are part of social structures and are influenced by external relations. This will result in consequential security requirements for the information systems. Examples are: anonymity, accountability, proof of a transaction, privacy, proof of proper functioning and legal proof.
- Security requirements regarding openness:  
An open system may consist of many elements (hardware, networks, operating systems, databases and other applications). It is required that each element of an open system (called an open element) is able to offer security in co-ordination with other open elements.

For some of the initiatives a table is used to summarise the ability of an initiative to fulfil the requirements as stated above. The layout of these summarising tables is shown in table 1 (the terms horizontal and vertical security are explained in the next section). Note that these tables are not intended to be used independently of the text of the report.

Table 1: Layout of summarising tables

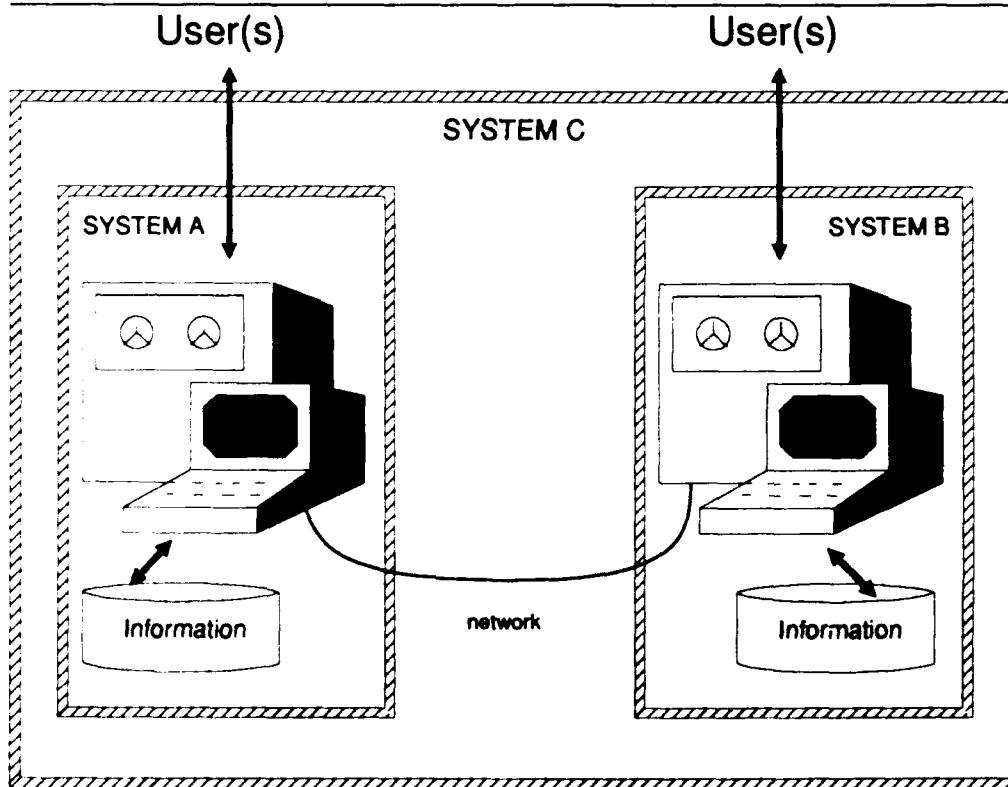


## 3

## A MODEL OF SYSTEMS IN A NETWORK

In order to present a structured inventory of the current initiatives in the area of secure open systems, a model of a real information technology (IT) environment is used. In figure 2, two computers are shown that are able to communicate with each other using a network.

It is reasonable to assume that services are provided by each computer in such a way that it performs as an autonomous information processing environment, thus we can speak of a system (system A and system B in figure 2). In this report, such a system is called a *computer system*. On the other hand, when services are distributed among more computers in the network, these



The figure shows three systems. Systems A and B are computer systems. System C is a networked system and encompasses systems A and B. The figure shows that it is primarily the chosen viewpoint regarding services and information that defines the borderline of a system.

Figure 2: Model of systems in a network

services, together with the associated operating systems, computer hardware, networks, etcetera, form an autonomous information processing environment, also creating a system (system C in figure 2). In this report, such a system is called a *networked system*. Thus, it is primarily the chosen viewpoint regarding services and information that defines the borderline of what belongs to a system and what not.

In figure 3 the distribution of information and services between and in systems is shown. This model is based on [67, 65] and is using the same approach as the ISO OSI-model [12]. (Human) users have access to applications. Applications offer the services that present and give access to information for the users. The applications have access to the information via the operating system. Also, communication with other applications takes place via the operating system. The operating system hides configuration and manufacturer-dependent characteristics from the applications (and the users). The abstraction level is different from that of the applications in the sense that the operating system only handles 'structured data' without knowing its meaning. The network is one of the configuration-specific characteristics that is hidden by the operating system. The network can be seen as a shared configuration-specific element of the connected operating systems.

Communication between applications is based upon a peer-to-peer relation. The security of this communication must also be based on this peer-to-peer relation. These peer-to-peer relations also exist between communicating operating systems and between communicating network entities. Security based on peer-to-peer relations is called *horizontal security*, since it is dealing with the same level of abstraction.

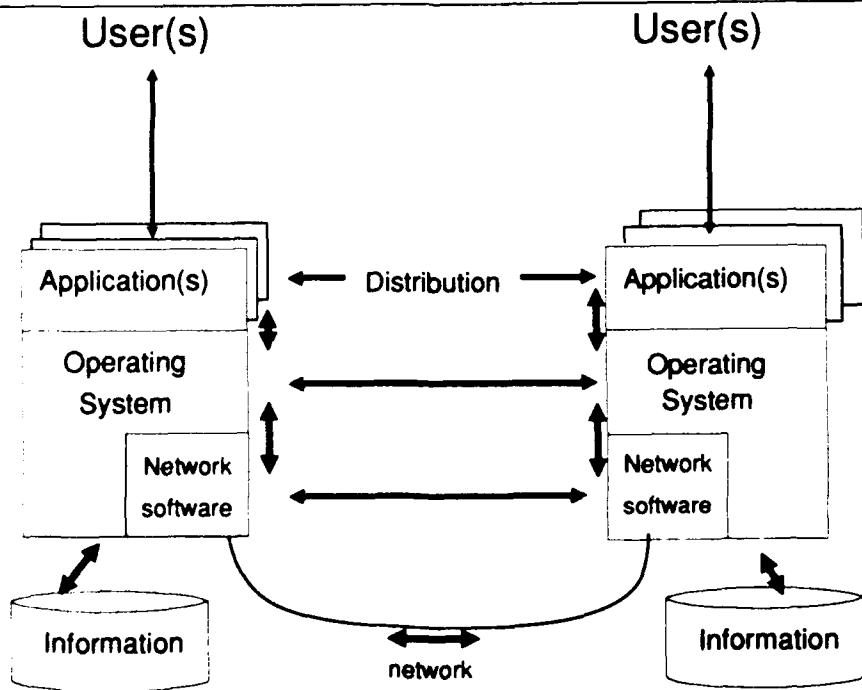
Generally, there is no *direct* communication between peers. Two applications can only exchange information through the operating system and devices, possibly using the network. So, the real communication takes place through several layers. The security that is needed to secure the communication between layers is called *vertical security*. *Vertical security* is a prerequisite for *horizontal security*.

Note that, whatever form the technical security takes, some minimal physical security will always be needed.

Horizontal security implies security between communicating applications, as well as security between communicating operating systems and security between communicating network entities.

Vertical security implies security between application and operating system, as well as security between operating system and network.

Security is always based on assumptions about the trustworthiness of the elements in the system (what elements have to be trusted, or what elements can be trusted). In the next sections we will investigate which of the initiatives take these trust relationships into account.



The horizontal arrows represent communication between peers at the same abstraction level. The communication concerns distribution of information and/or services. Except at the lowest level there is no direct communication between the peers. The actual route of the communication flow is denoted by the vertical arrows and the one horizontal arrow at the lowest level.

Figure 3: Distribution in a network

## 4

## INITIATIVES IN THE AREA OF SECURE OPEN SYSTEMS

Figure 4 is an abstraction of figure 3. For the purpose of this investigation the following major areas are recognised in which initiatives in the field of open systems security take place: applications, operating systems, networks and 'products' (a combination of hardware, software, etcetera, together a system). We will use this model in order to present a structured inventarisation of the current initiatives in the area of secure open systems.

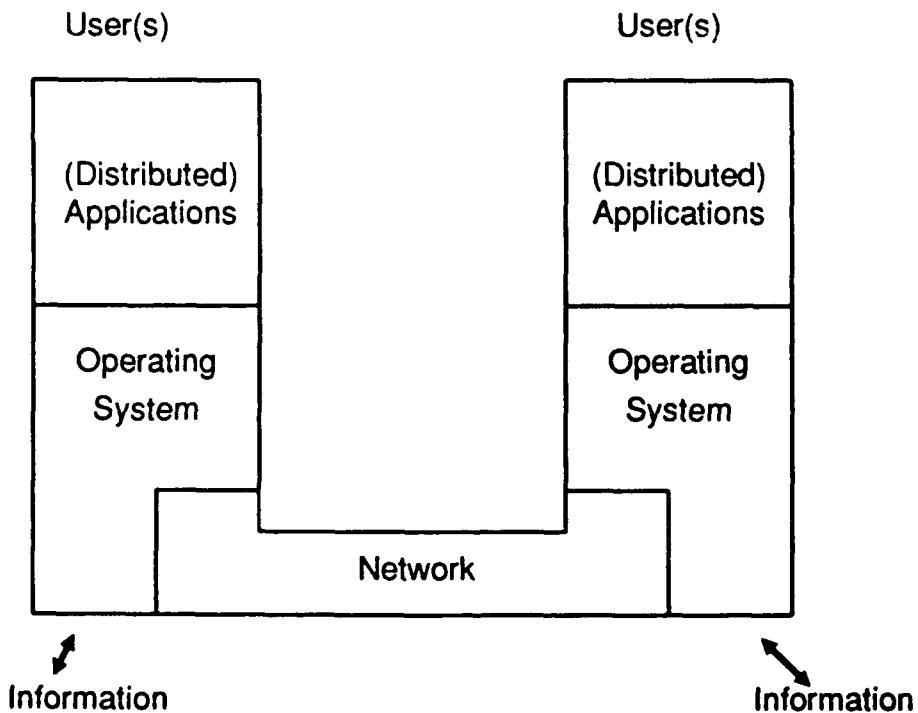


Figure 4: Areas of security initiatives

For the purpose of this investigation a considerable amount of documents was studied. First an inventory of ongoing standardisation activities in the area of security took place. Many sources were used for this inventory, most notably [52, 53 and 10]. Next, all documents that offered or supported (part of) an architecture for security in open systems were studied, varying from provisional drafts to definite international standards published before January 1992.

Surprisingly, we found that there is a tremendous amount of effort going on in the area of IT security. Nevertheless, there is only a limited number of activities that aim, directly or indirectly, at standardisation of security in open systems. These are briefly introduced in the following section. The initiatives, in as far as they are not generalised, are grouped in the following areas: applications, operating systems and networks.

#### 4.1 Applications

In the area of security offered by applications, more or less in cooperation with the operating system, many initiatives are taking place. The most important initiatives are:

- 1 The Framework for Secure Open Systems is produced by the ECMA [46]. This framework addresses the requirements and concepts for the provision of security in open distributed systems. Although the approach is suitable to be applied more generally, the security services primarily focus on the applications.
- 2 Recommendations meeting the design, specification and support of distributed applications in open systems are prepared by the CCITT Subgroup VII/Q19 programme. One of the activities is the development of a framework for security in distributed applications [4].
- 3 The Trusted Database Interpretation (TDI) [55] of the Trusted Computer System Security Evaluation Criteria (TCSEC) is developed by the American National Computer Security Center (NCSC). The TDI focuses on security in applications in general and database management systems in particular.
- 4 Security in OSI applications is primarily addressed by the ISO and the CCITT. Applications like Electronic Data Interchange (EDI) [7], The Directory (X.500) [16] and Message Handling Systems (MHS) [37, 38] provide interfaces to add security services at a later stage.

#### 4.2 Operating systems

- 1 Evaluation criteria for security in operating systems are defined in the Trusted Computer System Evaluation Criteria (TCSEC) [6]. The TCSEC was developed by the American Department of Defense / National Computer Security Center (DoD / NCSC). The TCSEC contains different sets of evaluation criteria for security which in practice work as design criteria. It has had, and still has, a tremendous impact on the security of operating systems.

- 2 The development of the Portable Operating System Interface for Computer Environments (POSIX) [49] is supported by the Institute of Electrical and Electronics Engineers (IEEE). The POSIX initiative aims at defining a standard interface set for applications. This interface set is to be offered by the operating system. The purpose of the Security Interface of POSIX is to define a standard interface for applications that require a secure environment.

#### 4.3 Networks

- 1 Most important for security in networks is the Open Systems Interconnection (OSI) Security Architecture [13]. It describes security services, mechanisms and the recommended placement of these within the OSI layers.
- 2 NATO OSI Security Architecture (NOSA) [39] is NATO's unclassified version of the OSI Security Architecture.
- 3 For security in networks the NCSC developed the Trusted Network Interpretation (TNI) of the TCSEC [56].
- 4 As a result of the Massachusetts Institute of Technology (MIT) Athena project, Kerberos was developed [35, 36]. Kerberos is an authentication service for users, end systems (computer systems) and applications in networks.

#### 4.4 General

- 1 Four European countries, France, Germany, The Netherlands and the United Kingdom, are harmonising criteria for the evaluation of security in information technology products. The result of this effort is the Information Technology Security Evaluation Criteria (ITSEC), a framework for the evaluation of technical security [30].

#### 4.5 Other initiatives

Other initiatives have also been studied, see the list below. These initiatives either not address technical security at an architectural level or use an approach that was derived or adopted from one of the initiatives mentioned above.

- ISO Joint Technical Committee 1 (JTC1), Sub Committee 27 (SC27) "Security Techniques" is the youngest sub-committee of ISO, established in 1990. At the time of writing (December 1991), the scope of work of SC27 was determined and the working program of SC27 is more or less stable [51]. SC27 has not yet produced any results of its own that need to be considered in this report. It is likely that SC27 will be an important and influencing party in defining standards for security in the future.
- ISO JTC1 SC21 "Information Retrieval, Transfer and Management for OSI" has several security standards under development, most notably the Security Models and Frameworks [19-24, 54]. These are based on the OSI Security Architecture.
- Another product of ISO JTC1 SC21 is the OSI Management standard [14]. Among other aspects, OSI Management addresses the management aspects of security, e.g. the management of security services and mechanisms. OSI Management does not add security services. However, the security in the OSI environment also depends on proper management and thus on OSI Management. On the other hand, the security of OSI Management depends on the security in the OSI environment. Therefore, OSI Management is briefly introduced in section 7.1, 'OSI Security Architecture'.
- ISO JTC1 SC6 "Telecommunications and exchange between systems" is responsible for the lower layers of the OSI model. SC6 has developed some security standards [28, 29]. These are based on the OSI Security Architecture.
- ISO Technical Committee (TC) 68 "Banking and Related Financial Services" defined several standards that enable trusted communication between banks [1, 2]. These standards address the exchange of cryptographic keys, cryptographic mechanisms and operational procedures for the use of cryptographic applications in a banking environment. Many of the TC68-standards stem from American National Standards Institute (ANSI) standards and Federal Information Processing Standard Publications (FIPS PUBs).
- ISO JTC1 SC18 "Text and Office Systems" (TOS) has defined the Office Document Architecture (ODA) [17]. Addendum 4 [18], currently a draft, addresses the format and structure of a document that also contains protected parts. The definition of the protection, e.g. by cryptographic techniques, falls outside the scope of ODA addendum 4.
- The IEEE 802.10 program "Standard for Interoperable LAN Security" (SILS) is addressing security issues in LANs with end systems like PCs and small workstations that do not necessarily implement all OSI layers [50]. SILS aims at offering security services at the lower OSI layers, especially layer 2 (data link layer).

- Both the Open Software Foundation (OSF) and the X/Open group are dedicated to the creation of internationally supported, vendor-independent software based on *de facto* standards to encourage the development of open systems. Both organisations base security on selected initiatives listed in the previous sections [45, 48].
- The Open Implementors Workshop (OIW) aims at defining profiles, including profiles for security in X.400, Directory, OSI Management and OSI lower layers. OIW's work is based on OSI standards.
- The Independent European Programme Group (IEPG), together with ECMA, aim at the development of so-called integrated project support environments (IPSEs). An IPSE is an environment supporting software engineering projects. The Portable Common Tool Environment (PCTE+) is an interface set on which a set of tools for software engineering can be built that together create an IPSE [33, 41]. PCTE+ also defines some interfaces to support security. The security functionality that is available using the PCTE+ interface set primarily addresses the security requirements in a software engineering environment. In addition, the PCTE+ interfaces give access to a selection of TCSEC security functions. Since PCTE+'s field of application is exclusive to the software engineering environment it is not further discussed in this report.
- Also studied were the activities of the European Telecommunications Standards Institute (ETSI), the European Workshop for Open Systems (EWOS) and the Information Technology Advisory Expert Group for Information Security (ITAEGV). Currently, these efforts, as far as relevant in the scope of this study, are based on initiatives listed in the previous section.

#### 4.6 Map of initiatives: studied aspects

In the next sections the standardisation efforts that have been introduced above are studied with respect to the security requirements addressed and their approach to technical security. Section 5 addresses initiatives in the application area, section 6 describes initiatives in the area of security in operating systems, section 7 discusses network security and finally section 8 discusses an initiative that aims at systems as a whole.

To facilitate a comparison, each of the identified initiatives is studied using the same list:

- 1 A short description of the background of the initiative.
- 2 A description of the approach towards security, the security architecture and the security services that are offered, as far as applicable.
- 3 Openness: the relationship with other security-providing parts in the system, especially (other) applications, the operating system and the network. Does the studied initiative result in an independent solution to security? Are other security measures assumed to be in effect, e.g. physical security? Will it offer services to, and/or require services from other elements of the system? What assumptions are being made about physical or organisational security? These questions address the issues of distribution of trust between the elements of an open system as well as the horizontal and vertical security.
- 4 Which of the remaining security requirements of section 2.5 are addressed and which are not?

#### 4.7 Guide to the readers

Readers that are not familiar with security standards are suggested to read some sections in alternate order:

- Section 6.1 before section 5.3.
- Section 7.1 before section 5.2.

Readers that only require a brief introduction to the most important initiatives for security are suggested to read at least sections 5.1, 6.1, 7.1 and 8.1.

Readers that are interested in an historic overview of the evolvement of security standards may decide to read the sections in the following order:

1985

- Publication of the TCSEC, section 6.1

1987

- Publication of the TNI, section 7.3

1988

- Final version of the OSI Security Architecture, section 7.1
- Publication of the ECMA Framework, section 5.1
- First steps in the development of Kerberos, section 7.4
- Publication of the NATO version of the OSI Security Architecture, section 7.2

1989

- First draft of the CCITT Support Framework for Security in Distributed Applications, section 5.2

1990

- First important external draft of the POSIX Security Interface, section 6.2

1991

- Publication of the ITSEC (frozen for a two-year trial period), section 8.1
- Publication of the TDI for an trial period of at least one year, section 5.3

## 5

## INITIATIVES THAT ADDRESS SECURITY IN APPLICATIONS

## 5.1 ECMA Framework for Secure Open Systems

## 5.1.1 Background

The ECMA published its Framework for Secure Open Systems in 1988 [46]. It uses a building block approach with which secure applications may be constructed. The major objectives in the development of the ECMA Framework were:

- to allow effective interworking of diverse products,
- to allow modular, expandable development of products,
- to accelerate the development of secure applications.

It is notable that these objectives directly address 'openness'.

One of the reasons for the development of this Framework was ECMA's experience that the criteria of the TCSEC (see 6.1) are focussed on the needs of military users and hardly address the needs of civil users.

Although the ECMA Framework can be applied in a more general way, it concentrates on the applications and, in a network, the application layer functions (ECMA intends to conform to the OSI model). The reason not to enlarge this scope is that each ECMA member, all of which are computer manufacturers, has its own specific interests in e.g. operating systems and networks. Based on the ECMA Framework two supplementary documents are offered: "Data Elements and Service Definitions" [47] defines data structures and services to support the ECMA Framework [46] and "Security Application - Authentication and Privilege Attribute" [43] defines attributes and details the data structures for authentication and privileges for secure applications.

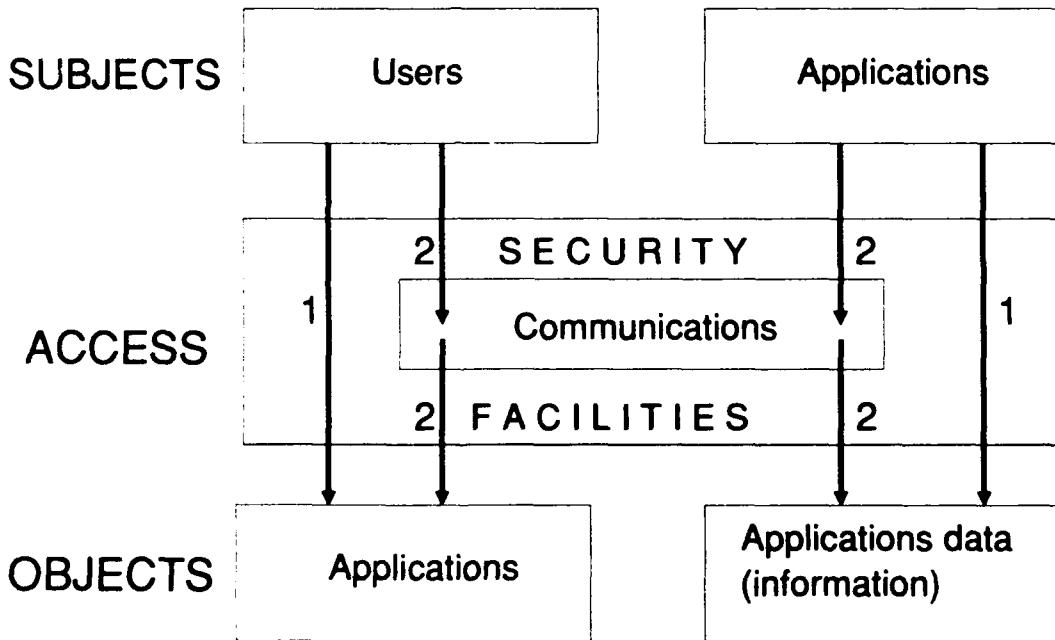
## 5.1.2 Architecture and services

Security in the definition of the ECMA Framework for Secure Open Systems is the resistance of *data processing systems* (information systems) to attack and misuse.

The ECMA Framework identifies the following threats: disclosure of information, contamination of information, unauthorised use of resources, misuse of resources, unauthorised information flow and denial of service.

Although identified, threats to availability like denial of service lie outside the scope of the ECMA Framework: ECMA concentrates on confidentiality and integrity.

Figure 5 shows the placement of security functionality in the ECMA Framework. It shows that



**Subject:** An active entity that initiates an action causing information flow or access to applications (or services).

**Object:** An entity that is (in the process of) being accessed.

Access of a subject to an object is always mediated by the ECMA Security Facilities.

**Arrows 1:** Direct access of a subject to an object, e.g. a user has access to an application or an application has access to information.

**Arrows 2:** Access of a subject to a remote object via communication facilities. The ECMA Security Facilities are invoked twice: first when a subject requests access to the communication facilities and the second time when the communication facilities request access to an object.

Figure 5: Conceptual view of the placement of the ECMA Security Facilities

every access of a subject to an object is always mediated by the Security Facilities, even when a remote object is accessed using the communication facility.

In the definition of the ECMA Framework, an *object* is: an entity in a passive role to which a security policy applies. And the ECMA definition for *subject* is: an entity in an active role to which a security policy applies.

Beside the well-known subjects and objects like (human) users, applications, network services and files (not all of them in the figure), the ECMA Framework recognises application-defined structures, the so-called *data\_objects*, as objects. The ECMA Framework does not dictate a security policy. Notable, however, is the presence of the Clark/Wilson-model [60]. The Clark/Wilson model describes access rules in the form of triplets: access control decisions are based on the *triplet* *USER/APPLICATION/DATA\_OBJECT*.

#### 5.1.2.1 The ECMA Domain concept

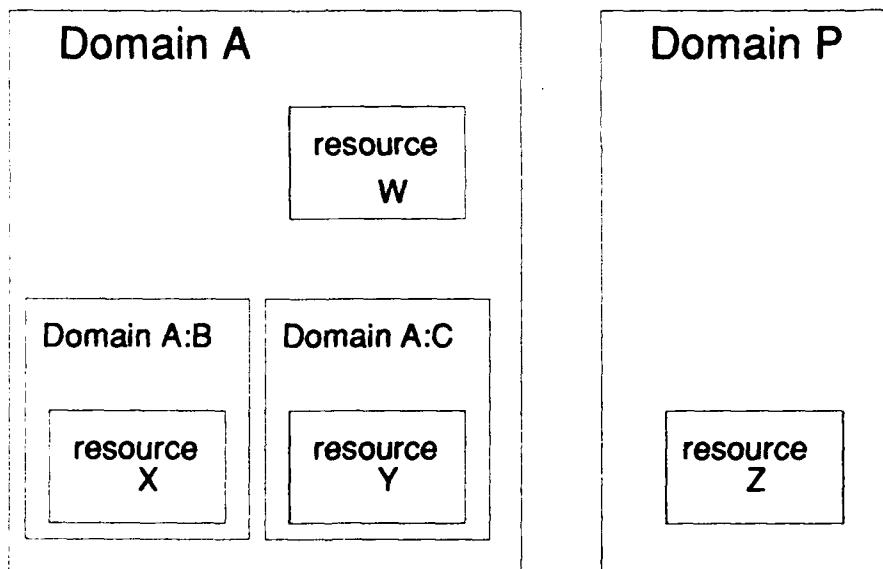
The definition in the ECMA Framework of a *Security Domain* is: a bounded group of objects and subjects (also called resources) to which a single security policy, executed by a single security administrator, applies. The Security Facilities enforce a security policy within a security domain. A domain does not relate to the organisation itself but to a *technically* bounded group. Examples of domains in the ECMA definition are: an application, an end system, a local network with connected end systems. E.g. domain A in figure 6 may be the end-system security domain and domain A:B may define the (local) security policy of an application. The security domains may be nested. The security facilities must be able to find out which security policy is to be applied. When inter-domain communication takes place, the policy of a shared superdomain is applied (e.g. domain A in figure 6 for communication between domain A:B and domain A:C). When no shared superdomain exists or domains are autonomous, a common denominator between domains will be based on negotiation between the peer domains (e.g. for communication between domain A and domain P in figure 6).

#### 5.1.2.2 The ECMA Security Facilities

The Security Facilities are building blocks in the provision of security services. The Security Facilities offer 'trusted functionality'. Yet, following the ECMA Framework, one of the goals in the security design of a system should be that security never entirely depends on one of the Facilities. The design should be such that a failure of a Facility will be compensated for (detected, repressed and possibly corrected) by the others. The ECMA Framework defines ten Security Facilities, briefly described below.

### Subject Sponsor

The Subject Sponsor is the intermediary between the subject and other security facilities. It is the only facility that directly communicates with the subject. The Subject Sponsor is aware of all ongoing activities of a subject. The Subject Sponsor is able to deal with subjects being human end users (see figure 5) as well as with applications and network services. So, the human end user is part of the ECMA model.



The notation A:B means that domain A:B is a subdomain of domain A.

- Use of resource X by a subject within domain A:B is controlled by the security policy of domain A:B.
- Use of resource Y by a subject within domain A:B is controlled by the security policy of the domain that is the least extensive superdomain embracing both domains A:B and A:C, which is domain A.
- Since no shared superdomain exists, use of resource Z by a subject within domain A:B will be based on negotiations between the peer domains A and P.

Figure 6: Security domains in the ECMA Framework

### Authentication Facility

This facility validates the identity of the subject. The subject can be a user or an application. Authentication takes place only once when a subject accesses the subject sponsor for the first time and starts a new association. The result of the authentication is passed to the Subject Sponsor in the form of a certificate that is signed by the Authentication Facility. This certificate will be used by the Subject Sponsor during the remainder of the association to meet all other requests for a 'proof of identity' as may be requested by other Security Facilities.

### Association Management Facility

This facility provides the management and enforcement of association security. It uses the Authorisation Facility (see below) for information about effective rights of the communicating entities and their authentication certificates. The Association Management Facility is also responsible for invoking the proper security mechanisms offering the required security for an association.

### Security State Facility

This facility preserves all *dynamic* security information within its security domain. All other Security Facilities consult the Security State Facility to get the necessary information of the security state in the domain and communicate the security-relevant results of their actions to this facility.

### Security Attribute Management Facility

This facility preserves all security information with a more or less *static* nature within a security domain. This facility provides for the management of security attributes (e.g. privileges and security controls) of subjects and objects.

### Authorisation Facility

This facility authorises or denies a requested access. Decisions are based on the access context (Security State Facility), the access privilege attributes of the subject and the access control attributes of the object (both sets of attributes are available through the Security Attribute Management Facility).

### Inter-Domain Facility

This facility maps one security domain's interpretation of security into another security domain's interpretation. This is especially challenging when no shared superdomain exists and communication takes place between domains with different 'authorities', e.g. different organisations. Note that this facility requires a (standardised) mutual 'security language' to exchange security information.

### Security Audit Facility

This facility provides for alarm and audit functions. It uses information that is gathered by other Facilities, especially the Secure State Facility.

### Security Recovery Facility

Real or suspected breaches of security that are detected by the Security Audit Facility cause predefined actions to be taken by the Security Recovery Facility.

### Cryptographic Support Facility

This facility provides cryptographic services for the other Facilities.

The management of these Security Facilities is briefly addressed as well as the way the management services interact with the Facilities. The ECMA documents [47] and [43] go into more detail with respect to some of these facilities. Work has started on defining the required interfaces, functionality and special protocols.

#### 5.1.2.3 ECMA Security Services

As shown in the description of the Facilities, the ECMA Framework does not directly address the identified threats to security. The Facilities are the building blocks for security services that address the identified threats directly (also see [47]). The ECMA anticipates the following security services:

- Access control, supported by authentication, access authorisation and inheritance functions<sup>2</sup>.
- Resource protection addresses integrity of resources, confidentiality of use of resources, assurance of service and accountability of use of resources.
- Information protection during processing, storage and interchange (interchange protection includes confidentiality, integrity and non-repudiation of interchange of information).
- Security Management includes administration, audit and event handling, e.g. recovery.

#### 5.1.2.4 Example of a Certificate

The ECMA Framework uses the concept of the certificate to exchange security information. A certificate is a 'token' with security information, for example a certain privilege. To be able to verify the integrity and the authenticity of the token, the contents is 'sealed' both with a cryptographic integrity code and the cryptographic signature of the 'Authority' that issued the certificate. Depending on the intended use of the certificate, different validity parameters can be added, e.g. addition of a 'validity time'-parameter to protect against play-back.

<sup>2</sup> Inheritance functions regulate the propagation of rights and duties in the system

One of the advantages of such a cryptographically protected certificate is that it can be given as a 'token' to a user or application; and that it does not have to be protected, e.g. in communication. An example of a data structure that contains privileges is the Privilege Attribute Certificate (PAC). For illustration, a PAC with some of the many possible fields is listed in table 2 (also see [47], [43] and [69]). A PAC may be 'to the bearer', like money, or 'in name', like a cheque. In the latter case, the identity of the PAC owner can be added to the PAC or the PAC can be embedded in or sealed with an Authentication Certificate that proves the identity of the PAC-owner.

Table 2: Example fields in the Privilege Attribute Certificate (PAC)

<u>Field Name</u>	<u>Description</u>
Privilege attributes	List of actual privileges
Validation key identifier	Encrypted key to protect against misuse
Validity time	
.	
.	
.	
. (other fields . with security . information)	
.	
PAC Authority	Signature of authority that issued and sealed this PAC
PAC Seal	Seal binding the contents

### 5.1.3 Openness and relations with other security-providing parts in the system

The ECMA Framework assumes that physical and procedural security measures are in effect. Furthermore, the security of the information in a security domain and the integrity of the Security Facilities therein depend on the protection by the superdomains (e.g. an Application Access Policy can easily be frustrated by a conflicting End System Access Policy). Conflicts like these must be solved using the Inter-domain Facility.

Three levels of domains are given as examples:

- 1 The Distributed System Security Domain. Its scope is the security of interoperability between end systems. This includes at least the network but may also include the connected end systems and/or distributed applications. The distributed-system policy may have areas of overlap with other policies, for example the security policy (or policies) of the end systems.
- 2 The End System Security Domain. Its scope is the individual end system, including applications, operating system and hardware.

3 The Application Security Domain. Its scope is a given application. The application may be distributed, e.g. a database system with clients and servers. In that case, the application domain has areas of overlap with many end system domains. Also, as more applications may reside in one end system, several application security domains may reside in one end system.

It is clear that security domains depend on one another. It is even possible that security conflicts between different security domains may arise. At this moment ECMA has not yet further defined the Inter-domain Facility. It is this Facility that should be dealing with the issue of distribution of trust between domains.

The granularity of protection fully depends on the domain structure. For example, the smallest information unit may be a field in a record, under the control of an Application Security Policy and the entire information base in an organisation may depend on the Distributed System Security Policy.

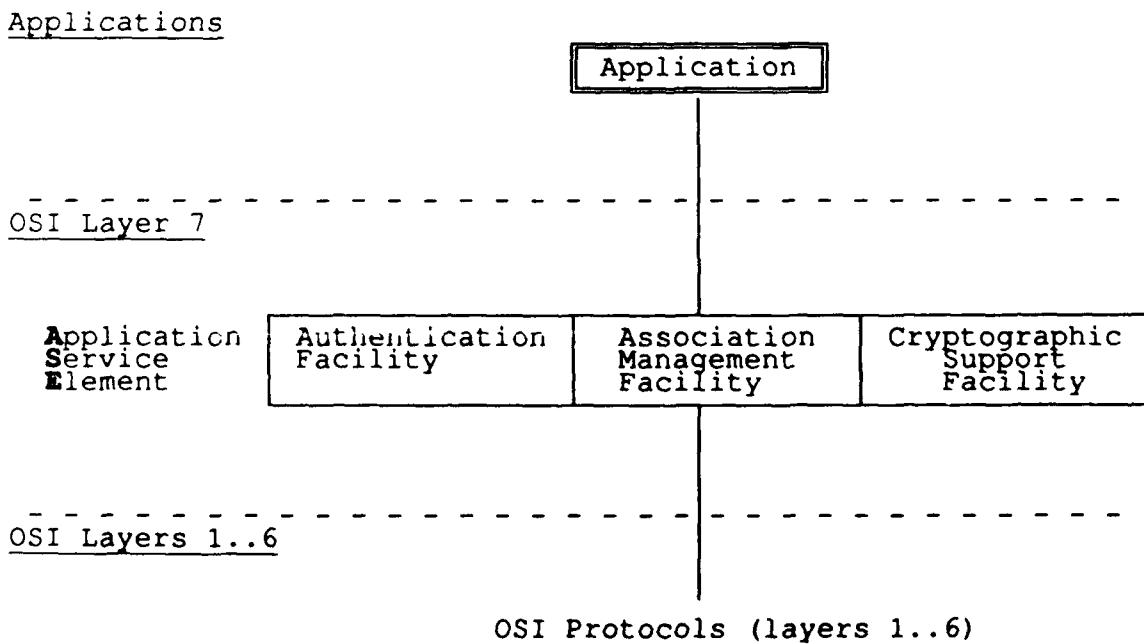


Figure 7: Security Facilities in an OSI Application Service Element

In case 'communications' (see figure 5) take place via an OSI network, the Security Facilities can be used as building blocks in the OSI Supportive Security Applications (SSA) or the OSI Application Service Elements (ASE). These service elements are also referred to as Security ASEs. In this case, the Facilities are to be offered by or via the Application Layer (see figure 7). Note that the application communicates with the network directly, outside the control of the operating system.

In ECMA document [47] ECMA seems to have given priority to a Distributed System Security Domain (see page 33) for applications. In this case, both the provision of security at the end systems (to be provided by the operating systems of the end systems) and the availability of the security services from the OSI security architecture are assumed to be available.

It can be concluded that the ECMA Framework is aware of the problems of security that are specific to open systems. Much depends on the structure of the domains and the Inter-domain Facility. The mechanism of issuing credentials that are cryptographically sealed enables the distribution of trust and can offer a basis for horizontal security. Vertical security fully depends on the domain structure and the availability of a suitable Inter-domain Facility between the domains.

#### 5.1.4 Which security requirements are addressed

Being the nature of a framework, it is clear that the ECMA Security Framework for Security in Open Systems only is the first step to security. All Security Facilities need specific protocols and a finer granularity of service definition. Keeping this in mind, the following list summarises whether the remaining requirements of section 2.5, can possibly be fulfilled or not.

##### Representation of real-life tasks and responsibilities in the system

Representation of real-life tasks is not specifically addressed although separation of duties is mentioned in passing (the 'controlled' should not have 'control'). A special role in the system is the role of the Security Administrator, which corresponds with a function in the organisation: the person(s) that is/are responsible for implementing the security policy in a security domain.

##### Responsibilities can be modelled as rights for the triplets

USERS/APPLICATIONS/DATA\_OBJECTS, possibly with additional use of sensitivity labels to support a Bell - LaPadula policy (see section 6.1).

**(Granularity of) mapping real-life tasks to services**

Only limited by the granularity of the security domain hierarchy.

**(Granularity of) mapping responsibilities to rights and duties to services and information**

The ECMA Framework both offers privileges for subjects and control attributes for objects.

The granularity depends on the security domain hierarchy. The control attributes can also be used to exclude specific access types or exclude access by identified subjects.

**Authentication information**

The ECMA Framework defines the Authentication Facility that issues Authentication Certificates and a Security State Facility. There is no restriction to *what* can be authenticated: (human) users, applications, services, end systems, etc.

**Security information (rights/duties)**

Two Security Facilities preserve and give access to the security information: the Security State Facility and the Security Attribute Management Facility. Both facilities have not yet been thoroughly defined.

Management of security information is only addressed briefly.

**Availability of services and information**

Availability is not a target for the ECMA Framework, although it is identified as a security requirement.

**Confidentiality of services and information**

Confidentiality is identified by the ECMA Framework as a security requirement. The ECMA Framework claims that confidentiality services can and will be provided based on the ECMA Framework. This requires specific protocols and functionality (e.g. cryptographic algorithms) that still need to be developed.

**Integrity of services and information**

Integrity is identified as a security requirement. ECMA claims that integrity services can be provided based on the ECMA Framework. Development of specific protocols and functionality is needed.

**Prevention**

Prevention of security events is implicitly addressed. Emphasis is put on prevention of loss of confidentiality and integrity.

**Reduction**

Reductive security measures are not addressed.

**Detection**

Audit and security alarms are services of the Audit Facility.

#### **Repression**

**Repressive security measures are not addressed.**

#### **Correction**

**The Security Recovery Facility will deal with correction.**

#### **Evaluation**

**Not addressed.**

#### **Mutual trust between users and system**

**The users have no means of verifying proper operation within a security domain. A security domain may either enforce trust on the user (by controlling all security-relevant actions or limiting the scope of the whole of his actions) or depend on another security domain to do so.**

#### **Distribution of trust in the system**

**When more than one security domain is involved, the distribution of trust must be achieved via the Inter-domain Facility. This Facility is not yet sufficiently defined.**

**Within one security domain this distribution exists by definition. The ECMA Framework uses encrypted certificates as the basic mechanism for the distribution of trust.**

#### **Trust relations between the elements of the system**

**The security domains may hierarchically depend on one another. It is not clear how ECMA intends to provide a trust enabling service between domains. The proper placement for such a service is in the Inter-domain Facility.**

#### **Requirements for security imposed by society**

**Not directly addressed. The building block approach does not exclude the provision of additional services to fulfil these requirements.**

**The possibility to fulfil security requirements using the ECMA Framework is listed in table 3.**

Table 3: Security requirements addressed in the ECMA Framework for Security in Open Systems

<u>Requirement</u>	<u>Addressed?</u>
Real-life tasks -> Services	No
Responsibilities	Privileges for subjects
Duties/obligations	Security controls for objects
Exclusions	Yes, based on security controls
Control of use of services	Yes, granularity: depends on domain
Control access to information	Yes, granularity: depends on domain
Authentication	Yes: (human) users, services, etc.
Security information	Yes, specific Security Facilities
Security management	Identified as a requirement, not a target
Confidentiality	Needs development of specific protocols
Integrity	Needs development of specific protocols
Availability	Identified as a requirement, not a target
Prevention	Yes
Reduction	Not addressed
Detection	Audit and alarm
Repression	Not addressed
Correction	Yes, Recovery Facility
Evaluation	Not addressed
Mutual trust	Not addressed
Requirements from society	Not directly addressed
Focus on	Applications
Aware of	Networks
Horizontal security	Inter-domain Facility between domains
Vertical security	Certificates within a domain
Distribution of trust	Depends on domain structure
Trust relations	Inter-domain Facility
	Yes, between domains

## 5.2 CCITT Distributed Applications in Open Systems (DAF)

### 5.2.1 Background

The CCITT is currently developing the Support Framework for Distributed Applications (DAF) [8], which is to become a multi-part standard with one of its parts dedicated to security (DAF Security, [4]). This Support Framework aims to support the development of distributed applications in OSI networks.

The DAF Security document is a draft and not yet stable. For this reason DAF Security will only be introduced briefly. Nevertheless, the general architecture is clear. This architecture is influenced mostly by the OSI Security Architecture (see section 6.1) and by the ECMA Framework for Security in Open Systems (see section 5.1).

### 5.2.2 Architecture and services

The three central themes in DAF-Security are Trust, Security Policy, and Security Domain. Trust is needed in a Security Domain to be able to enforce a Security Policy within this domain. The use of the terms Security Policy and Domain is similar to that in the ECMA documents. Trust is defined as the confidence that an entity to which trust is applied will perform in a way which will not prejudice the security of cooperating (trusting) entities. It is expected that 'trust' may not only be invoked by technical means, but may also require physical security.

#### 5.2.2.1 Threats

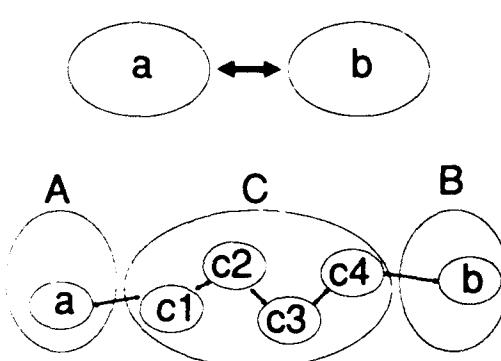


Figure 8:  
DAF communication scenario

The composition of processes, services, applications, etcetera, that together perform as a whole is called the Behavioural Component (BC). When two BCs communicate with one another, their view of the communication is that of the model shown in the upper part of figure 8. Actually, the lower part of the figure presents a better representation of the interaction between the BCs. With this representation in mind, DAF identifies the following threats: masquerading, manipulation, repudiation, data interception (traffic analysis and other means of information gathering), information sequencing (includes relay, replay, delay and prelay<sup>3</sup>), denial of service, identity interception (loss of anonymity), unauthorised access (after authentication) and object reuse.

#### 5.2.2.2 The DAF Security Model

DAF Security Services are provided by Security Facilities. Each application consists of one or more processes and application data. Distributed applications may consist of processes running at different end systems and connected by an OSI network.

Any BC which includes the provision of security facilities is classified as a Security Component (SC). SCs may be decomposed recursively, until an atomic level is reached at which the required degree of trust can be established (this may require a module that is secured *physically*). BCs

<sup>3</sup> Prelay is used as the opposite of delay; e.g. sending a premature message

providing security facilities may be composite in nature and comprise other BCs, some of which may be BCs themselves. The security policy governing an SC will determine the set of rules for protecting BCs associated with the SC. Again, the policy may prescribe the use of physical security.

#### 5.2.2.3 Services

DAF-Security defines *communication-related Security Facilities* and *management-related Security Facilities*.

Possible types of communication-related Security Facilities are:

- authentication
- integrity,
- non-repudiation,
- anonymity,
- access control,
- authorisation,
- data confidentiality,
- cryptographic support,
- traffic padding,
- routing control.

Definitions will most likely be taken from the OSI Security Architecture and the ECMA documents. Notable is the definition of an anonymity service which is unknown in the OSI SA or the ECMA Framework.

Possible types of management-related Security Facilities are:

- security audit,
- notarisation<sup>4</sup>,
- key management,
- registration of names and identities,
- management of security attributes (grant, distribute and revoke rights, privileges, etcetera).

Similar to the ECMA Framework, the basic mechanism in the provision of security will probably be the 'certificate'.

<sup>4</sup> The notarisation service provides for registration of security-relevant events by a trusted third party.

### 5.2.3 Openness and relations with other security-providing parts in the system

The question that is raised by figure 8 is where the trust is situated. Can BCs 'a' and 'b' trust their own environments 'A' and 'B'? Do they have means of verifying that? Are the two BCs *aware* of the existence of 'C', and, if so, who is the proper authority to verify 'C'?

Currently, it is not clear whether DAF-security is going to solve this question at all. At least, DAF-security seems to be dependent on the security services of the OSI network. It is likely that there will also be a dependency on the security that is offered by the operating system. Finally, DAF-security explicitly states that the trust issue may mean that some physical security is required.

### 5.2.4 Which security requirements are addressed

The DAF-security document is still a non-stable draft version. Therefore, this section is left blank.

## 5.3 Trusted Database Management System Interpretation of the TCSEC (TDI)

### 5.3.1 Background

The Trusted Database Management System Interpretation (TDI) of the Trusted Computer System Evaluation Criteria (TCSEC) focuses on security in applications in general and database management systems (DBMSs) in particular. The TDI is intended to be used in conjunction with the TCSEC itself (see section 6.1). Readers unfamiliar with the TCSEC are suggested to read section 6.1 first, since the TDI is built upon the TCSEC. The TDI is published by the USA NCSC in April 1991 for an trial period of at least one year. The TDI is also known as the Grey Book, after the colour of the cover.

Unlike other publications of the NCSC, the TDI was produced with extensive community review and is heavily influenced by database manufacturers.

There are some inconsistencies in the current version of the TDI. One example: as the document stands now, it would be possible that a TCB, regarded as a composition of TCB-subsets, will be evaluated successfully whereas the same TCB, regarded as a whole, would fail. This discussion lies outside the scope of this study.

### 5.3.2 Architecture and services

The TDI has the same objectives as the TCSEC but extends these objectives beyond the scope of solely the operating system. The TDI addresses the combination of the applications and the operating system in a system. The reason is clear: applications, specifically DBMSs have their

own datastructures with a much finer granularity than the operating system. The TDI offers protection to the level of records and fields in a file, whereas the TCSEC-protection is (in practice) limited to a file as a whole (see page 51).

As in the TCSEC, the basic mechanism for security in the TDI is the Trusted Computing Base (TCB) of which the reference monitor mediates all access attempts to a set of objects. These sets are clearly identified. In addition to the TCSEC, the TDI introduces *TCB subsets*.

Definition: a *TCB subset* is a set of software, firmware and hardware (where any of these three could be absent) that mediates every access of a set of subjects to a set of objects.

The TCB subset uses resources that are provided by an explicit set of more primitive TCB subsets. The TCB subset uses these resources to create and manage its set of objects, as well as for its own internal needs. The most primitive TCB subset has an explicit set of resources.

The TDI is using the concept of the *trusted subject*. Definition: a *trusted subject* is a subject that is permitted to have simultaneous read and write access to objects of more than one sensitivity level. Examples of objects with more than one sensitivity level are: a database file that contains records with different sensitivity levels; a harddisk that contains files with different sensitivity levels. These objects are composed of several other objects with a finer granularity. A trusted subject implements a Reference Monitor mechanism with respect to the objects it is permitted to access. This creates the basis for a less primitive TCB subset. The less primitive TCB subset is protected by the more primitive TCB subset (the most primitive TCB subset may be protected by non-technical measures). The less primitive TCB subset as well as its set of objects is composed of resources and objects out of the set of the more primitive subset.

The combination of all TCB subsets in a system must create one TCB for the whole system.

Although the TDI can be applied more generally, the most common situation will be that the more primitive TCB subset will be implemented by the operating system and less primitive TCB subsets will be implemented by applications (which are the trusted subjects). Each of the less primitive TCB subsets will be protected and separated from one another by the more primitive TCB. The more primitive TCB subset will provide the less primitive TCB subsets with their own resources (which are objects of the more primitive TCB). Within these resources, a less primitive subsets creates and manages its own objects.

**Example:**

A system consists of two TCB subsets. The first is the more primitive TCB subset, implemented by the operating system. The second is the less primitive TCB subset, implemented by a database management system (DBMS).

This operating system's TCB subset protects access to the following objects:

- a file which is, when executed, the DBMS,
- this DBMS in its processing environment (so, the DBMS's TCB subset is protected by the operating system's TCB subset),
- a database file,
- other files, processes, etcetera.

The operating system's TCB has been installed in such a way that the DBMS is a trusted subject with respect to all accesses to the database file. The operating system's TCB will see to it that only the DBMS will be given access to the database file. No other subjects may have direct access to the database file.

The DBMS implements the less primitive TCB subset. The DBMS's TCB protects access to the following:

- records, fields, tables, relations, tuples and all other elements within the database file (these are all objects within the DBMS's TCB subset),
- structures and resources that are internal to the DBMS's TCB subset.

### 5.3.3 Openness and relations with other security-providing parts in the system

One of the benefits of the TDI architecture is that the evaluation of security can be done in parts, that is, within certain conditions. It is possible to compose a system of an operating system (that implements the most primitive TCB subset) and applications (that all implement their own less primitive TCB subset). These elements may already have been evaluated and the evaluation results can be reused. In that case 'only' the interactions and dependencies in the composite system have to be evaluated. The most important trust relations (from a technical point of view) between the hierarchical TCB subsets are:

- uniform handling of subject-sensitivity labels,
- global identification and authentication,
- trusted path,
- audit.

- a system architecture that offers:
  - protection of less primitive TCB subsets,
  - separation of less primitive TCB subsets,
  - protection of the objects that *may make up a trusted subject*,
  - protection of those (composite) objects and/or resources that enclose a TCB subset's objects,
- correspondence between TCB subsets' security policies (including access control policies),
- consistency of TCB subset interfaces.

Since these trust relations effect the security of the composition of TCB subsets as a whole, they are called *global requirements*, as opposite to *local requirements* that only effect the security within a TCB subset.

The TDI recognises that the security of a less primitive TCB subset depends on the more primitive TCB subset. As such, it is a good example of vertical security. The most primitive TCB subset may be protected by non-technical security measures. The security of the composed TCB depends on all the TCB subsets individually as well as the combined effort of these TCB subsets. The TDI is written with the relationship of applications and the operating system in mind. The concept of TCB subsets enables the local provision of security in the operating system or in an application and offers a promising mechanism for vertical security. If properly defined TCB subset interfaces were available, it would be possible to develop operating systems and applications, each implementing its own TCB subset independently of other TCB subsets. This would be a true benefit to security in open systems. The TDI does not solve the problem of the TCB subset interfaces. The current effort of the NCSC is in demonstrating the concepts of TCB subsets based on combinations of specific DBMSs and operating systems (only the complete package can be offered for evaluation).

Can the TDI approach be extended to the network? No, regrettably it is not. The reasons are:

- Firstly, the network cannot be a *more* primitive subset than the operating system because the network software depends on the operating system and the network is being resourced by the operating system.
- Secondly, the network cannot be a *less* primitive subset than the operating system because:
  - it depends on many operating systems' TCB subsets,
  - it is not protected by one TCB subset and
  - its objects are not embraced in one TCB subset's objects.

In both cases, there is a problem regarding the required partial ordering of the more/less primitive TCB subsets. It is clear that an operating system not (only) depends on the network. On the other hand, the network does not depend on one operating system either.

Thus, we conclude that it is not easy to extend the TDI approach to networked systems.

#### 5.3.4 Which security requirements are addressed

By definition, the TDI addresses the same security requirements as the TCSEC (see section 6.1.4). The only difference is that the TDI extends the security functionality to the applications. Thus, not only the objects that can individually be managed at the operating system level are protected, but also those at the application level. This enables a much finer granularity of protection. On the other hand, the TDI has the same limitations of the TCSEC and ignores security requirements that are specific to applications or, more specific, DBMSs. These requirements are not found in the TCSEC, take for instance integrity.

The following list summarises whether the remaining requirements of section 2.5 can be fulfilled with the functionality as described in the TDI (in conjunction with the TCSEC) or not. Since the TDI is directly built upon the TCSEC we will only list the differences and refer to the TCSEC for details.

##### Representation of real-life tasks and responsibilities in the system

See TCSEC.

##### (Granularity of) mapping real-life tasks to services

See TCSEC.

##### (Granularity of) mapping responsibilities to rights and duties to services and information.

See TCSEC.

##### Authentication information

See TCSEC. The TCB subsets have to work together to uniquely identify and authenticate users as well as to associate identified users to auditable events.

##### Security information (rights/duties)

See TCSEC. Each TCB subset has a subset Reference Monitor that may have its own Reference Monitor database. Consistency between these databases (e.g. the labelling principle) must somehow be achieved.

##### Availability of services and information

See TCSEC.

#### Confidentiality of services and information

See TCSEC. The granularity of protection can be much finer, up to the level of records and fields.

#### Integrity of services and information

See TCSEC.

#### Prevention

See TCSEC.

#### Reduction

See TCSEC.

#### Detection

See TCSEC. The TCB subsets must work together to provide a system-wide audit.

#### Repression

See TCSEC.

#### Correction

See TCSEC.

#### Evaluation

See TCSEC.

#### Mutual trust between users and system

See TCSEC.

#### Distribution of trust in the system

Yes, but of a static nature. Trust is situated in each of the TCB subsets. Each TCB subset has its own set of objects to protect. A TCB subset can be installed in such a way that a trusted subject (and only that specific trusted subject) will have unlimited access to a set of objects under the control of the TCB subset. This trusted subject will implement a less primitive TCB subset.

#### Trust relations between the elements of the system

Less primitive TCB subsets depend on more primitive subsets. The security of the whole TCB depends on all individual TCB subsets as well as the combined effort of the TCB subset.

#### Requirements for security imposed by society:

See TCSEC.

The fulfilment of requirements, as far as they are different from the TCSEC, is listed in table 4.

Table 4: Security requirements addressed in the TDI (and different from TCSEC)

Requirement	Addressed?
<u>Exclusions</u>	<u>Yes, only trusted subject has access to certain objects</u>
Control of use of services	Yes, granularity: depends on TCB subset (no limitation)
Control access to information	Yes, granularity: depends on TCB subset (no limitation)
Security information	Several TCB subset Reference Monitor databases
Focus on	Applications and Operating systems
Horizontal security	No
Vertical security	Yes, between TCB subsets
Distribution of trust	Yes, trust in each TCB subset
Trust relations	Yes, between TCB subsets in partial order

## 5.4 Application-dependent security: FTAM, EDI, MHS and The Directory

### 5.4.1 Background

In section 5.2 the question was raised to what extent the applications must be able to trust the environment in which they run and on which they depend. In some applications, this dependency is intentionally reduced to the bare minimum (not only for security reasons but also for increased portability). Examples of standards for these applications are [1] and [2]. These standards are primarily intended for use in the banking environment. An overview of these standardisation efforts can be found in [34].

Some standards offer hooks to add security. Such a hook is an interface definition without a service specification. Standardly, the resulting call will be empty. Manufacturers and users can add security functionality afterwards. Examples are File Transfer, Access and Management (FTAM) [15], EDI [7] and Message Handling Systems (MHS) [37, 38]. It must be noted that these security hooks were added to the standards at a later stage of their development and are not mandatory. Moreover, the placement of the hooks restricts the possible security services that can be made available through these hooks (see also [62]).

The current initiatives in the area of security in applications are poorly coordinated. For example, in [10] we counted 16 *different* methods for authentication in applications.

#### 5.4.1.1 The Directory

Of particular interest is the Authentication Framework of The Directory [16], also known as X.509. The Directory can be seen as the distributed 'phone book' in an OSI environment. The

Directory is typically used to facilitate communication between, with and about objects such as OSI application-entities, individuals, devices and distribution lists. The Authentication Framework defines a framework for the provision of authentication services to its users. For this, the Directory contains security information like 'credentials' and public keys. Two levels of authentication are defined: simple authentication using a password as a verification of the claimed identity and strong authentication involving credentials based on public key cryptographic techniques.

This framework for authentication is mentioned here because it is gaining a wide acceptance and because the same concepts are used by other applications. There also is some criticism concerning the limitations of the current approach [61]. These limitations currently are under discussion in the ISO [25]. One of the major problems of the Directory is to maintain the integrity of the stored information, especially of the encryption keys. This discussion lies outside the scope of this report.

## 6

## INITIATIVES THAT ADDRESS SECURITY IN OPERATING SYSTEMS

## 6.1 Trusted Computer System Evaluation Criteria (TCSEC)

## 6.1.1 Background

The NCSC, which is part of the DoD, published the Trusted Computer System Evaluation Criteria (TCSEC) in 1983 [5]. The TCSEC is better known as the Orange Book because of the colour of the cover. A revised edition was published in 1985 [6].

The TCSEC has three objectives:

- 1 To provide *guidance to manufacturers* as to what to build into their new, commercial off-the-shelf trusted IT products in order to satisfy trust requirements for sensitive applications and as a standard for DoD evaluation thereof.
- 2 To provide users with a *yardstick* with which to assess the degree of trust that can be placed in computer systems for the secure processing of classified or other sensitive information.
- 3 To provide a basis for specifying *security requirements* in acquisition specifications.

Although the criteria are intended for general use, the scope of the TCSEC is in practice restricted to non-networked (stand-alone) systems with emphasis on the operating system. 'Interpretations' of the TCSEC are necessary to deal with specific security feature requirements, e.g. for networking, databases and embedded systems.

At the time of publication the TCSEC was the only standard for evaluation of security in IT products. Furthermore, the DoD required that new systems are evaluated against the TCSEC criteria. For these reasons, the impact of the TCSEC on computer security was, and still is, immense.

## 6.1.2 Architecture and services

The security architecture in the TCSEC is based on three concepts: the reference monitor, a formal security policy model and the trusted computing base (TCB). These concepts are briefly described below.

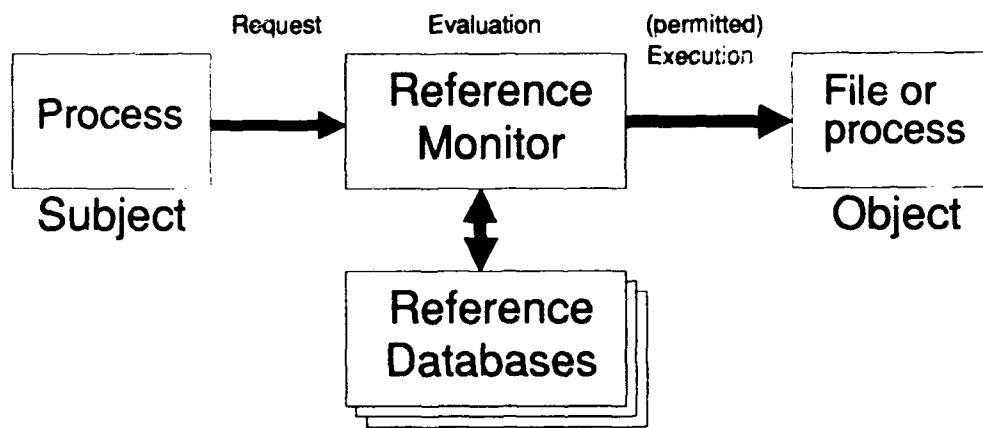


Figure 9: The Reference Monitor

#### 6.1.2.1 The reference monitor

The idea behind the reference monitor is simple (also see [66]): every request for a security-relevant action in the system is evaluated before its (possible) execution. The initiating entity of a request is called the *subject* (a process acting on behalf of either a user, another subject or the system), the addressed entity is called the *object* (e.g. a workstation, a file, a data element or another process).

The official TCSEC definitions are:

*Subject:* An active entity, generally in the form of a person, process, or device that causes information to flow among objects or changes the system state. Technically it is a process/domain pair (domain: see below).

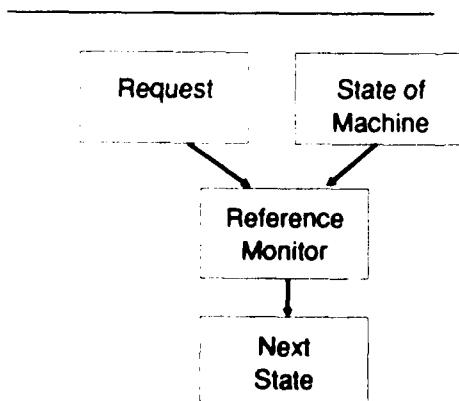
*Object:* A passive entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects are records, blocks, pages, segments, files, directories, directory trees, and programs, as well as bits, bytes, words, fields, processors, video displays, keyboards, clocks, printers, network nodes, etcetera.

*Domain:* The set of objects that a subject or resource in an automated information system has the ability to gain access.

Note that the area that is depicted in the definition of *object* is much broader than the area that is actually covered by the TCSEC. For example, network and database security lie outside the scope of the TCSEC.

The reference monitor guarantees controlled state transitions from a safe state to another safe state. See figure 9. The evaluations of the reference monitor are based on static security information (e.g. rights of the user or process; requirements for the access of a certain file or other process) and dynamic security information (e.g. other processes working on the same object; specific system management taking place; time of day; history).

The security information is available in security reference databases. Modification of this security information is under the control of the reference monitor itself. The reference monitor is responsible both for the decision and the enforcement of that decision.



The set of objects in the system under control of one reference monitor is the *domain* of that reference monitor. Any object is controlled by precisely one reference monitor. During the evaluation of a request, no security-relevant changes inside the domain of the reference monitor are possible. The situation is frozen, thus creating a single-state machine (figure 10).

Figure 10: Single-State Machine

The main characteristics of the reference monitor can be summarised as follows:

- 1 The reference monitor creates a single-state machine that only allows transitions from one secure state to another.
- 2 The reference monitor mediates all changes within its domain. During the evaluation of a request the situation in the domain is frozen.
- 3 The security information that is needed by the reference monitor is under its control, including changes to this information.

#### 6.1.2.2 A formal security policy model

The reference monitor enforces the security policy within its domain. The TCSEC models security according to the Bell - LaPadula policy and adopted this policy as its formal security policy [57]. The two rules of the Bell - LaPadula policy are:

- 1 A subject can *read* an object only if the hierarchical classification<sup>5</sup> in the subject's security level is greater than or equal to the hierarchical classification in the object's security level and if the non-hierarchical categories in the subject's security level include all the non-hierarchical categories in the object's security level (Basic Security Theorem). This clause is often referred to as 'no read up'.

In a semi-formal way:

Read Access is permitted if-and-only-if

LEVEL OF SUBJECT is greater or equal to LEVEL OF OBJECT

AND

CATEGORIES OF SUBJECT include *all* CATEGORIES OF OBJECT

- 2 A subject can *write* an object only if the hierarchical classification in the subject's security level is less than or equal to the hierarchical classification in the object's security level and all the non-hierarchical categories in the subject's security level are included in the non-hierarchical categories in the object's security level (this is called the star-Property, or \*-Property). This clause is often referred to as 'no write down'.

<sup>5</sup> Note:

Hierarchical classifications are, for example, Unclassified, Confidential, Secret, Top Secret.

Non-hierarchical categories are, for example, organisational units like departments or organisational structures like project groups.

In a semi-formal way:

Write Access is permitted if-and-only-if

LEVEL OF SUBJECT is less or equal to LEVEL OF OBJECT

AND

*all* CATEGORIES OF SUBJECT are included in CATEGORIES OF OBJECT

Example:

Facts:

- User PAUL's clearance is Confidential.
- User PAUL is a member of the following non-hierarchical categories: Research-group-26, Research-group-32, Project SEDIS.
- File TEXT has the classification Secret.
- File TEXT belongs to the non-hierarchical categories Research-group-32, Project SEDIS.

1 Rule 1 of the Bell - LaPadula policy says:

Read Access of PAUL to TEXT is permitted if-and-only-if

CLEARANCE OF PAUL is greater or equal to CLASSIFICATION OF TEXT

*(This is false since Confidential is less than Secret)*

AND

CATEGORIES OF PAUL include *all* CATEGORIES OF TEXT

*(This is true since Paul's set {Research-group-26, Research-group-32, Project SEDIS} includes the TEXT's set {Research-group-32, Project SEDIS}).*

Thus, read access is denied for lack of clearance.

2 Rule 2 of the Bell - LaPadula policy says:

Write Access of PAUL to TEXT is permitted if-and-only-if

CLEARANCE OF PAUL is less or equal to CLASSIFICATION OF TEXT

*(This is true since Confidential is indeed less than Secret)*

AND

*all* CATEGORIES OF PAUL are included in CATEGORIES OF TEXT

*(This is false since from Paul's set {Research-group-26, Research-group-32, Project SEDIS} element Research-group-26 is missing in the TEXT's set {Research-group-32, Project SEDIS}).*

Thus, write access is denied because TEXT does not belong to Research-group-26.

#### 6.1.2.3 The Trusted Computing Base (TCB)

The kernel of the trusted computer system is the TCB. The TCB contains all elements of the system responsible for supporting the security policy and the isolation of objects on which the protection is based (most prominent: the reference monitor and its security reference databases). Or, in other words: the TCB contains all security-enforcing functions. Outside the TCB there are no elements that need to be trusted to maintain protection.

#### 6.1.2.4 Security functionality

Based on the architecture that is described above the TCSEC offers the following security functionality, called *control objectives*:

##### Security Policy

The Bell - LaPadula policy (only) addresses confidentiality. The enforcement of this policy takes place by discretionary and/or mandatory access control when access to an object is requested. To assist the functions that enforce access control, the TCSEC may require the clearing of information in objects before reuse as well as sensitivity labelling of all objects.

##### Accountability

Accountability in the TCSEC definition is the possibility to keep an individual responsible for his actions (blame-ability). Accountability in the TCSEC only addresses access to (sensitive or classified) information or actions that influence the possibilities for such an access. Accountability of use of resources is not a target. To enable accountability, the TCSEC requires identification and authentication of users as well as audit of the so-called 'security-relevant' actions that may cause creation of, access to, or effect the release of classified or sensitive information.

##### Assurance

Assurance concerns itself with guaranteeing or providing confidence that the security policy is correctly implemented and that the TCB does indeed accurately mediate and enforce the intent for that policy. Evaluation of assurance includes issues concerning the design, development and maintenance of systems, the possibilities for evaluation and testing, and the operational security.

#### 6.1.3 Openness and relations with other security-providing parts in the system

The relations with other security-providing parts of the system are determined by the boundary of the TCB. All elements inside the TCB are protected by the TCB and are assumed to offer trusted functionality (as far as within the scope of the TCSEC security policy).

What is protected by the TCB depends on the granularity of the Reference Monitor. Most often *inside* the TCB we find: storage devices (disk and tape units, memory, CPU, 'bus'), device drivers, queues, the filing system, the 'kernel' of the operating system (including the security-enforcing functions that create the TCB). Note that the hardware in this list is assumed to be protected by physical measures.

*Outside* the TCB are in general: Applications, networks, terminals and storage media.

All elements that are outside the TCB are considered to be insecure. So, there are no trust relations with elements outside the TCB. The boundaries of the TCB are static. The TCB is considered as a whole (although it is not strictly forbidden, it is almost impossible to get a TCSEC-evaluation of a composed TCB). The substitution or removal of any part of the TCB, should it be possible at all, invalidates a TCSEC evaluation. The TCSEC approach gives many problems in open systems since open systems are of a composite nature.

In practice, the granularity of the TCB is a file or a process. When a finer granularity is needed than that of a file or a process this must be offered by other (non-TCB) security-enforcing functions in the system. Information and services that are outside the TCB must/will also depend on other (non-TCB) security-enforcing functions in the system. This especially is the case for applications and networks.

These additional security-enforcing functions offer functionality that lies outside the scope of the TCB. These functions are not part of the TCB. For their own security (e.g. their integrity), these functions must be able to trust the TCB on which they depend to a large extent.

#### 6.1.4 Which security requirements are addressed

The following list summarises whether the remaining requirements of section 2.5 can be fulfilled with the functionality as described in the TCSEC or not.

##### Representation of real-life tasks and responsibilities in the system

After authentication, users are represented in the system by processes acting on their behalf. The TCSEC is orientated towards the military environment. The only way to influence responsibilities of users is via their hierarchical clearance and non-hierarchical grouping in relation to that of the objects. This results in access rights to objects. In most implementations, the granularity of this protection is a process as the subject and a file as the object. The (only) method for mapping the real world to the system is by using the tools that

implement the Formal Security Policy (Bell - LaPadula). This model restricts accesses that might result in an information flow and is based on hierarchical and non-hierarchical characteristics of both subjects (users, programs) and objects (files, programs).

**(Granularity of) mapping real-life tasks to services**

Barely addressed. In the higher TCSEC classes, separation of the tasks of operators, administrators and auditors is required. This offers a first rudiment of separation of duties.

**(Granularity of) mapping responsibilities to rights and duties to services and information.**

Not addressed. After authentication all users can execute all services within the system.

**Authentication information**

The TCB requires users to identify themselves to the TCB before performing any of the actions that the TCB is expected to mediate. The TCB uses a protected mechanism (e.g. passwords) for authentication. The TCSEC only addresses *user* authentication.

**Security information (rights/duties)**

Security information is stored in the conceptual Reference Monitor database. Access to this security information also is under the control of the reference monitor. Management of security information is not addressed.

**Availability of services and information**

Not addressed

**Confidentiality of services and information**

Confidentiality of information is offered through the tools that implement the Formal Security Policy (Bell - LaPadula).

Confidentiality of services is not addressed.

**Integrity of services and information**

Barely addressed. In the higher TCSEC classes it is required that tools must be provided for the verification of the correct operation of the hardware and firmware elements of the TCB at the initial state of the computer system.

**Prevention**

Only prevention of loss of confidentiality.

**Reduction**

Not addressed.

**Detection**

Only detection of possible loss of confidentiality through audit. There is no requirement for real-time detection of security breaches. Detection may be based on an afterwards analysis of the audit files.

**Repression**

Not addressed.

**Correction**

Hardly addressed. In the higher classes 'procedures and/or mechanisms' for trusted recovery shall be provided.

**Evaluation**

Not addressed.

**Mutual trust between users and system**

The system does not have to trust the user (once properly authenticated) since the TCB checks every security-relevant action. The user does not have this possibility and has no way to convince himself of the proper behaviour of the system. In the higher TCSEC classes a so-called 'trusted login' is required. This 'trusted login' guarantees a trusted path between user and TCB for authentication purposes.

**Distribution of trust in the system**

All the trust is situated in the TCB.

**Trust relations between the elements of the system**

Outside the TCB there are no elements that need to be trusted (except physical security).

There is only one TCB. Systems are evaluated as a whole, also when they are of a composite nature.

**Requirements for security imposed by society:**

Not a target of the TCSEC.

A summarising overview of the fulfilment of requirements by the TCSEC is listed in table 5.

Table 5: Security requirements addressed in the TCSEC

<u>Requirement</u>	<u>Addressed?</u>
Real-life tasks -> Services	No
Responsibilities	Hardly
Duties/obligations	No
Exclusions	No
Control of use of services	Yes, granularity: whole system
Control access to information	Yes, granularity: file
Authentication	Yes, only users
Security information	Reference Monitor database
Security management	No
Confidentiality	Yes
Integrity	No
Availability	No
Prevention	Yes access control
Reduction	No
Detection	Partly through audit
Repression	No
Correction	Hardly
Evaluation	No
Mutual trust	No
Requirements from society	Not addressed
Focus on	Operating systems
Horizontal security	No
Vertical security	No
Distribution of trust	No, all trust is in the one TCB
Trust relations	No, all trust is in the one TCB

## 6.2 POSIX Security Interface

### 6.2.1 Background

The goal of the Portable Operating System Interface for Computer Environments (POSIX) is to define a standard operating system interface and environment based on the concepts of the UNIX Operating System to support application portability [11, 42]. Although based on UNIX, POSIX is intended to be independent of a specific operating system: many operating systems should be able to offer the POSIX interface sets and the required functionality.

POSIX is an initiative of the IEEE. POSIX defines interfaces and their functionality (not implementations) for applications to improve portability of source code (not the binaries). Both for commercial and historical reasons, POSIX does not specify all aspects of the interfaces to the operating system (e.g. administration and management are not defined). It allows vendor specific extensions at many places (which is a burden for one of the major goals: improved portability).

It was clear from the beginning of the POSIX initiative ( $\pm$  1984) that a security interface was needed. Work on the security interface actually started in 1988. The POSIX Security Interface is described in P1003.6 [49]. At the time of writing of this report P1003.6 was in its seventh draft version. The goal of the POSIX Security Interface is to specify the interface for additional system functions, modifications to non-secure system functions and commands for additional security within the POSIX system. In many places extensions and differentiations are permitted to provide greater security or fulfil the security needs of market areas with special needs.

The working group that is developing the POSIX Security Interface bases much of its work on the DoD / NCSC Trusted Computer System Evaluation Criteria (TCSEC, Orange Book), see section 6.1. The working group gradually becomes aware of other relevant standards and initiatives (see page 91 of [11]). This did not have any impact on the draft POSIX Security Interface so far.

An in-depth review of the POSIX Security Interface can be found in [68], on which this section is based.

### 6.2.2        Architecture and services

Figure 11 shows the placement of the POSIX interfaces, including the security interfaces situated between the applications and the functionality of the operating system. Entities that can be addressed through these interfaces are files, directories, devices and processes (including the so-called *pipes*). Thus, the granularity of protection is limited to a file or a process.

There are four possible routes to access the functionality of the operating system and access to information (see figure 11):

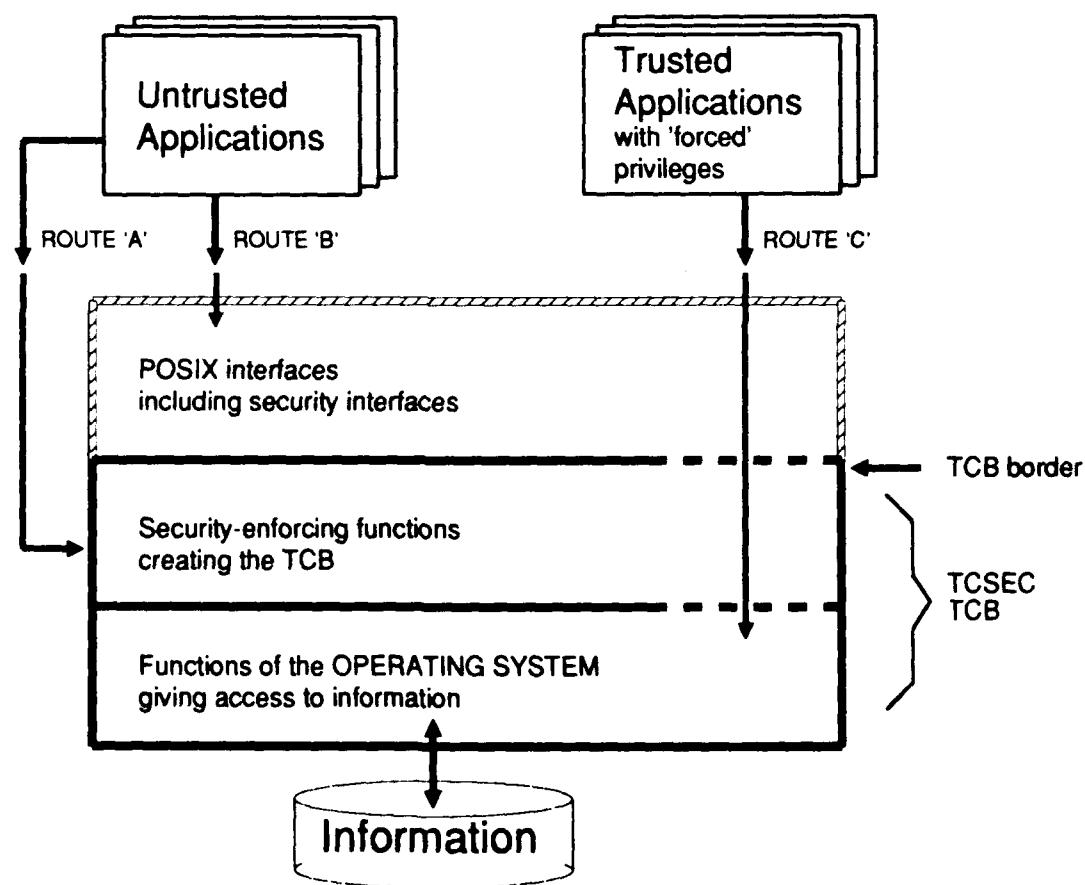
**Route 'A':** Applications that do not use POSIX at all or by-pass the POSIX Security Interface partly or completely. Applications that take route 'A' might use other interfaces or system libraries. Of course, these activities must be monitored as well, so they will be subject of mediation by (perhaps other security-enforcing parts of) the Trusted Computing Base (TCB). All security-enforcing functions together form the Reference Monitor and create the TCB (also see section 6.1).

**Route 'B':** Applications that use the POSIX Security Interface. The remainder of this section will concentrate on the security functionality that can be achieved via this route. The use of a specific security interface definition results in a 'call' of (implementation dependent) security functions that offer the defined security functionality.

**Route 'C':** The POSIX Security Interface offers the possibility to add a 'token' to an object that will force the granting of privileges once the object is executed (and becomes a subject). This granting of privileges is *not* mediated by security-enforcing functions of the TCB and is

not dependent on the user. The granting of the 'forced privilege'-token itself must be subject to mediation of the TCB (however, see note).

Route 'D' (not in figure 11): Security of access via a network and security of network services are not addressed by the current POSIX Security Interface.



Route 'A': Direct access to operating system functions, by-passing the POSIX (security) interfaces.

Route 'B': Access to operating system functions via the POSIX security interfaces and mediated by TCB enforcing functions.

Route 'C': Access to operating system functions via the POSIX security interfaces but by-passing parts of the TCB enforcing functions since 'forced' privileges are in possession.

Route 'D': (not shown in this figure) Access via a network.

Figure 11: Placement of POSIX and POSIX Security interfaces

Note: Application developers and users are free in their choice of one of these routes. The use of the POSIX interfaces is discretionary. It is, for example, always possible to use route 'A'. Therefore, at all routes the security functionality must be sufficiently rich to implement at least non-contradicting coherent security policies, which, if possible at all, is a burden for the management of security.

#### 6.2.2.1 Services

In P1003.6, an own security terminology is used. P1003.6 recognises the following security services (named 'security policies'): non-disclosure, integrity and accountability. These are discussed in some more detail below.

To support these services, the following mechanisms are anticipated:

- access control (discretionary and mandatory),
- audit,
- privileges,
- information labelling,
- object import and export (not relevant within the scope of this section).

##### Non-disclosure

Non-disclosure (confidentiality of information) is based on the Bell - LaPadula policy (see page 52). A more generalised form of the Bell - LaPadula policy, named *Information Label Policy*, is supported too. The main difference is that the need for a hierarchical scheme is relaxed. This policy is best illustrated by the following (imaginary) interface definition:

REQUEST (FROM SUBJECT 'A' WITH LABEL 'L1';  
FOR OBJECT 'B' WITH LABEL 'L2';  
FOR ACCESS 'READ')

Using the Bell - LaPadula policy this request would be evaluated by applying the rules for no-read-up/no-write-down. Read-access would be granted if 'L1'  $\geq$  'L2' and subject 'A' has read-access permission for object 'B'. Within the information labelling policy it is the system administrator that defines the outcome of the evaluation 'L1' in relation to 'L2'.

The non-disclosure service is the responsibility of the access control mechanisms, which use security-information like privileges and labels. It must be mentioned here that main areas on which the non-disclosure service depends are not addressed at all by POSIX, e.g. identification, authentication, ownership and authorisation are missing.

### Integrity

Although recognised as a required security service, the support for an integrity service is limited to grant or deny write-access to objects by setting the access control attributes to *read-only*. All other limitations for write-access are in support of the non-disclosure service.

### Accountability

Accountability in P1003.6, like in the TCSEC, aims at the possibility to keep a person responsible for his actions (blame-ability) as far as these actions might affect the non-disclosure policy. This service is offered by auditing mechanisms that allow for the collection and afterwards analysis of security relevant events on a per-user basis. Unlike the TCSEC, P1003.6 also offers the possibility for applications to add audit records to the audit files. It is not clear how accountability is feasible when applications take the route 'C' in figure 11.

It must be mentioned here that identification and authentication of users, essential to enable accountability, are not addressed by the POSIX Security Interface or elsewhere in POSIX.

#### 6.2.3 Openness and relations with other parts providing security in the system

The relations of the POSIX Security Interface with other security-providing parts of the system are determined by the functionality and granularity that is achievable through the interfaces and the binding of this functionality. The security interfaces do not offer security themselves and therefore do not have to be part of the TCB. One exception to this is the use of the 'forced privileges' where the security interfaces indeed do offer a possibility to by-pass the TCB.

Handling of the 'forced privileges' should therefore be part of the TCB.

Of course, the functionality of the POSIX environment depends on the integrity of the interfaces, which must be protected by the TCB.

### Application security

When a finer granularity of security is needed than that of a file or a process, this cannot be offered by the POSIX Security Interface. It should be provided by the application itself. The security in the application would depend on the security that is achievable through the POSIX Security Interface (e.g.: record security in the application, file security through the POSIX security interfaces).

One exception to this is the audit interface that enables the security functions in the application to use the central audit mechanisms to log auditable events in a format that can be specified by the application.

### Other routes (see figure 11)

As mentioned before, the use of specific routes is not mandatory. The possibility to use other interfaces or to access the functionality of the operating system directly will always exist. This means that access to information is possible in several ways. Therefore, a sufficiently rich security functionality must be offered to implement non-contradicting and coherent security policies at all routes.

It is difficult to imagine that different independent security-enforcing functions coexist for the different routes and still maintain one security policy. So, it is plausible that the different routes to the operating system and information will encounter the same security-enforcing functions of the operating system. This requires an interface definition of these security-enforcing functions. In POSIX terms: not only interface definitions for the applications are needed but also interface definitions for the TCB.

### Operating System security

POSIX security depends altogether on the security of the operating system. The operating system is assumed to be protected by the TCB which is part of the operating system itself.

### Network security

Network security is not addressed by the POSIX Security Interface. It is not clear whether that it is just ignored or it is assumed to be provided elsewhere. In both cases, POSIX security depends on network security. However, POSIX has no knowledge or access to information about network security.

The POSIX Security Interface is situated between the operating system and the applications. As such, it is the proper place to provide vertical security. Some of the security services of the operating system indeed are available to the applications through the POSIX Security Interface. The best example is the audit service. Horizontal security is not relevant in the scope of POSIX.

To summarise: POSIX security depends on the security functionality of the TCB, the security of other possible routes to the functionality of the operating system and the security of the network. When a finer granularity of protection than that of a file or a process is needed, it must be offered by the applications.

#### 6.2.4 Which security requirements are addressed

The following list summarises whether the remaining requirements of section 2.5 can be fulfilled with the functionality that is available through use of the POSIX Security Interface or not.

##### Representation of real-life tasks and responsibilities in the system

The POSIX Security Interface does not offer interfaces or functionality that enables the projection of the real-life tasks and responsibilities to rights and duties in the system. POSIX does not even have a clear perspective of what a user is. POSIX deals with user-identification numbers (UINs). These UINs do not necessarily relate to an identified person (several users may map to one UIN).

The limited tools for mapping responsibilities to the system support the Bell - LaPadula and the Information Labelling Policy.

##### (Granularity of) mapping real-life tasks to services

Not addressed.

##### (Granularity of) mapping responsibilities to rights and duties to services and information

The rights in the system do not directly relate to responsibilities in real life. Rights have to be modelled as privileges (although the mechanism is too restrictive for this). Duties are not addressed. The granularity of protection in the POSIX Security Interface is a process as a subject and a file or process as an object.

##### Authentication information

The POSIX Security Interface does not address identification and authentication. Access control decisions are based on the user-identification number (UIN) which is not controlled by POSIX and does not necessarily relate to an identified person.

##### Security information (rights/duties)

All security information is stored in and accessible by the TCB.

Management issues explicitly lie outside the scope of POSIX, including the management of security information.

##### Availability of services and information

Not addressed.

##### Confidentiality of services and information

Confidentiality of information (named non-disclosure) is offered by applying the Bell - LaPadula Policy or a system-defined Information Labelling Policy.

Confidentiality of services is not addressed.

**Integrity of services and information**

Hardly addressed. Although the need for integrity services is recognised, the only mechanism to support integrity is by making objects read-only.

**Prevention**

Only prevention of loss of confidentiality is addressed.

**Reduction**

Not addressed.

**Detection**

Only detection of possible loss of confidentiality through audit (accountability). This detection is not instantaneous but is available through 'off-line' analysis of the audit files.

Alarm functions are not defined.

**Repression**

Not addressed.

**Correction**

Not addressed.

**Evaluation**

Not addressed.

**Mutual trust between users and system**

The user has no means to verify the proper behaviour of the POSIX Security Interface.

An element in the system, solely using the POSIX Security Interface, has no means to verify the user since the POSIX Security Interface has no means for user authentication.

**Distribution of trust in the system**

The trust is situated in the TCB (handling of 'forced' privileges should be in the TCB as well). Network security and application security are not addressed, so no provisions for the exchange of security information (e.g. with the TCB) are given (except audit information).

**Trust relations between the elements of the system**

The POSIX Security Interface does not offer means either to the applications or to the users to verify the proper behaviour of the TCB or other security-providing parts in the system.

The TCB does not have to trust the POSIX interfaces since the TCB controls every security relevant action. Therefore, the POSIX Security Interface does not necessarily have to be part of the TCB. A potential exception to this is the handling of 'forced privileges'. As the document stands now, the handling of these 'forced privileges' may be done by the POSIX Security Interface. Since the handling of 'forced privileges' is a security-relevant action, this

must be done by the TCB. Thus, at least the parts of the security interface that handle these privileges should be part of the TCB.

Requirements for security imposed by society:

Not a target for POSIX.

A summarising overview of the fulfilment of requirements by the POSIX Security Interface is listed in table 6.

Table 6: Security requirements addressed in the POSIX Security Interface

<u>Requirement</u>	<u>Addressed?</u>
Real-life tasks -> Services	No
Responsibilities	Hardly: privileges based on user identification numbers
Duties/obligations	No
Exclusions	No
Control of use of services	Yes, same as TCB granularity: whole system
Control access to information	Yes, granularity: file
Authentication	No
Security information	In TCB
Security management	No
Confidentiality	Yes
Integrity	Hardly
Availability	No
Prevention	Yes, access control
Reduction	No
Detection	Partly through audit
Repression	No
Correction	No
Evaluation	No
Mutual trust	No
Requirements from society	No target
Focus on	Interface applications / operating system
Aware of	Applications and operating system
Horizontal security	(not applicable)
Vertical security	Yes
Distribution of trust	No, all trust is in the TCB
Trust relations	No, all trust is in the TCB

## 7

## INITIATIVES THAT ADDRESS SECURITY IN NETWORKS

## 7.1 OSI Security Architecture

## 7.1.1 Background

ISO identified the need for a series of standards to enhance security within the Open Systems Interconnection architecture. The most important one, and the starting point for all other security activities in the OSI environment, is the OSI Security Architecture (for short: OSI SA) which is an International Standard since 1988 [13]. Security in the OSI definition is 'minimising the vulnerabilities of assets and resources'. ISO 7498-2 extends the Basic Reference Model [12] to cover security aspects which are general architectural elements of communication protocols.

It is acknowledged that security in the OSI environment is only one of the aspects of information security [13; page 21]. For instance, security at the end systems lies outside the scope of OSI.

## 7.1.2 Architecture and services

OSI SA describes security services and related mechanisms and defines the position within the seven-layer OSI reference model where the services and mechanisms may be provided. OSI SA defines the following services:

## Authentication

The authentication service validates a claimed identity of (communicating) peer entities or the claimed network-addresses (sources) of information.

## Access control

The access control service must provide protection against unauthorised use of OSI resources.

## Confidentiality

OSI SA defines data confidentiality (protection of information against unauthorised disclosure), selected field confidentiality (aims at specific parts of a data-unit) and traffic flow confidentiality (the fact that communication takes place is protected against unauthorised disclosure).

### Integrity

The integrity service detects and prevents the acceptance of unauthorised alteration or deletion of data. The data integrity services partly depend on peer authentication.

### Non-repudiation

The non-repudiation services provide proof of delivery and proof of origin (the source) of a data unit. These services are partly based on peer authentication and can be considered as a special kind of integrity services.

The notarisation service is a special case of non-repudiation: a third party registers the activities of two communicating partners in the network.

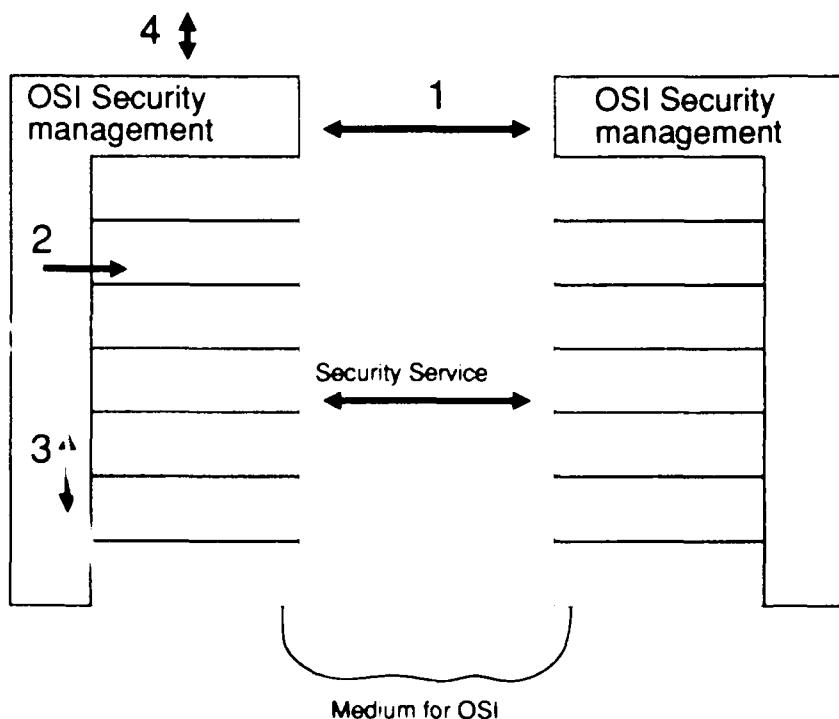
Most mechanisms are directly related to the services they support. Relations with mechanisms that lie outside the scope of the OSI SA are anticipated. For instance, trusted functionality is needed to maintain the integrity of the OSI environment: the end systems are to provide access control to OSI mechanisms and entities through means other than OSI services themselves (e.g. direct access through the operating system). For event detection (concerns security alarms), audit trails and recovery, relations with OSI management are anticipated [14].

#### 7.1.2.1 Security Management

The management of the security services is one of the tasks of OSI Security Management. The main responsibilities of OSI Security Management are briefly described below (see figure 12).

- 1 Management of the Security Management Information Base (SMIB). The SMIB is the conceptual repository for all security relevant information needed within a domain (defined in OSI as the set of end systems in a network that obey the same security policy). The SMIB is a distributed database. Each end system will have some security information locally stored to enable it to enforce the security policy. Security management may require the exchange of security relevant information between end systems in order to update and maintain consistency of the SMIB. Examples are: exchange of information about available security services, exchanges of security mechanism information (e.g. encryption keys), handling of events and audit.
- 2 Management of security services and mechanisms may consist of determination and assignment of a specified target of protection, negotiation and selection of mechanisms and key management.

- 3 Services may be offered through the combination of mechanisms at different layers. For this, exchange of security information between the layers is needed.
- 4 Relationships exist between security in the end system environment and management in the OSI environment. OSI management will be initiated at the end system. Furthermore, OSI management resides in the end system and is itself a managed object of the end system. Thus, the security of the OSI environment depends on the security of the end systems. This lies outside the scope of OSI.



This figure shows the seven-layer OSI structure embedded in the OSI management functions. The security management functions are:

- 1 Exchange and maintenance of security management information (SMIB),
- 2 Management of security services and mechanisms,
- 3 Inter-layer communications,
- 4 Communication with the world outside scope of OSI, which is the end system.

Figure 12: OSI Security Management

#### 7.1.3 Openness and relations with other security-providing parts in the system

For the security of the OSI environment, OSI SA assumes that physical and organisational security measures are taken. Furthermore, the integrity of the OSI-'stacks', the confidentiality of the stored security information in the local part of the SMIB and the availability of resources for the proper functioning of the OSI mechanisms depends on the local security management and the security of the end systems in which the OSI functions reside.

The granularity of protection by OSI SA-services typically is a data unit or all data units within an association. OSI is not aware of the local interpretation of data units (e.g. the data units may embody a file). Note that applications lie outside the scope of the seven-layer OSI model.

Remarkably, this includes OSI-defined standard applications like FTAM, X.400 and EDI (see section 5.4).

The security of the communicated data units fully depends on the security at the end systems (operating systems and applications). OSI has no knowledge of security at the end systems. It should also be noted that in some perceptions, the OSI environment is available for the applications directly, by-passing the operating system. In this view, the network is neither part of the operating system nor is it accessible (only) via the operating system. Therefore, in this view the applications must handle all the configuration specific network characteristics themselves, including those that address security.

Distribution of trust is an issue in the OSI SA. The OSI SA defines, for example, services for peer authentication and (peer) integrity. This also is an example of horizontal security. Vertical security is offered as well: OSI Security Management offers the possibility to exchange and manage security for inter-layer communication in the OSI environment.

#### 7.1.4 Which security requirements are addressed

The following list summarises whether the remaining requirements of section 2.5 can be fulfilled with the functionality as described in the OSI Security Architecture or not.

##### Representation of real-life tasks and responsibilities in the system

Outside the scope of OSI.

##### (Granularity of) mapping real-life tasks to services

Outside the scope of OSI.

##### (Granularity of) mapping responsibilities to rights and duties to services and information

Outside the scope of OSI.

#### Authentication information

The OSI SA defines an authentication service for the authentication of peer entities and (network)-addresses.

#### Security information (rights/duties)

All security information is stored in the SMIB.

OSI SA also defines that management services for the SMIB must be offered. This will probably be done in collaboration with the development of general OSI Management.

#### Availability of services and information

Not addressed.

#### Confidentiality of services and information

OSI SA defines several confidentiality services for the communicated data units. It defines a traffic flow confidentiality service as well.

#### Integrity of services and information

Several integrity services are defined.

#### Prevention

Prevention of loss of confidentiality and integrity is defined.

#### Reduction

One of the integrity services supports recovery for which reductive measures are defined.

#### Detection

Alarm reporting and audit are considered useful, but no services are defined. These services will be offered through OSI Management.

#### Repression

Although considered useful, no services are defined.

#### Correction

One of the integrity services supports recovery.

Recovery is a service in OSI Management as well, this is recovery from incidental errors.

#### Evaluation

Not addressed.

#### Mutual trust between users and system

Outside the scope of OSI SA.

### Distribution of trust in the system

OSI SA is using the concept of a distributed database that contains all relevant security information, called the SMIB. Based on the information in the SMIB, the security services provide distribution of trust as far as it lies within the scope of OSI. Two major problems remain:

- 1 The security of the SMIB depends on the security at *all* end systems since it is distributed and because partitions of the SMIB are located in all end systems;
- 2 The security depends on updating and maintaining of consistency of the SMIB (which is distributed and contains a lot of rapidly changing security information). The difficulty is to create a stable service to maintain the SMIB while this service depends on the SMIB for its own security at the same time.

### Trust relations between the elements of the system

The OSI environment fully depends on the security and proper management of *all* end systems in the domain. There is no possibility to verify from the OSI environment whether this is the case or not.

The users of the OSI environment (operating systems and applications) could verify the integrity of the OSI-'stack' as far as it is functioning locally at the end system (basically, this just would imply an integrity check on the local OSI software). There is no standard mechanism available to achieve this. Since the security in the OSI environment depends on the SMIB too (and thus on all other end systems) a local check is insufficient.

There are no technical means to enforce trust either from the network side or from the end systems' side. This implies that trust must be conceived by other, e.g. organisational, security measures.

In some views, the OSI environment is available for the applications directly, by-passing the operating system. Then, the applications are responsible for the provision of the required security. This means that the security of information at the end systems not only depends on the security that is offered by the operating system, but also depends on the security of the applications that use the OSI environment.

### Requirements for security imposed by society

Not directly addressed. The notarisation service, which is a special kind of non-repudiation service, may be implemented as a trusted third party and is able to provide proof of transactions and activities of partners in the network.

A summarising overview of the fulfilment of requirements by the OSI Security Architecture is listed in table 7.

Table 7: Security requirements addressed in the OSI Security architecture

<u>Requirement</u>	<u>Addressed?</u>
Real-life tasks -> Services	Outside scope of OSI SA
Rights/Responsibilities	Outside scope of OSI SA
Duties/obligations	Outside scope of OSI SA
Exclusions	Exclusion of peers
Control of use of services	Yes, network services
Control access to information	Yes, granularity: data unit
Authentication	Yes, of peers and network addresses
Security information	In SMIB
Security management	Yes, via OSI Management
Confidentiality	Yes
Integrity	Yes
Availability	No
Prevention	Yes
Reduction	Yes, partly
Detection	Partly through alarm and audit via OSI Management
Repression	Considered useful, no services defined
Correction	Yes, partly. Also via OSI Management
Evaluation	No
Mutual trust	Outside scope of OSI SA
Requirements from society	Notarisation service
Focus on	Networks
Aware of	Some applications
Horizontal security	Yes, between peers
Vertical security	Yes, between the layers and via OSI Management
Distribution of trust	Yes, distributed SMIB
Trust relations	Yes, many relations

## 7.2 NATO OSI Security Architecture (NOSA)

### 7.2.1 Background

NATO develops and uses its own standards and conventions. For data communications, a strategic decision was made to conform to the OSI standards (especially the seven-layer OSI Basic Reference Model [12]) as much as possible, see STANAG 4250 [40]. STANAG 4250 also profiles the options in the OSI model. In addition, the NATO OSI Security Architecture (NOSA) [39] was published in 1988 (unclassified). NOSA is a tailored equivalent of the OSI Security

Architecture. The OSI Security Architecture is insufficiently precise to permit NATO interoperability since it is too generic and contains too many options. NOSA has limited the placement of security services, mainly to layer 7 and to the layer 3/4-boundary. Note that there also exist a classified version of this standard which specifically addresses military requirements [44].

### 7.3 Trusted Network Interpretation of the TCSEC (TNI)

#### 7.3.1 Background

The DoD / NCSC published the 'Trusted Network Interpretation' (TNI) of the 'Trusted Computer System Evaluation Criteria' (TCSEC) in July '87 [56]. It is also known as the Red Book, after the colour of the cover. The TNI has the same objectives as the TCSEC (see section 6.1), but it is especially focussed on network security.

The TNI did not gain the same degree of acceptance as the TCSEC. At some places the TNI lacks the concreteness of the TCSEC and leaves many issues open for interpretation. Some of the issues raised seem to go beyond the state of current technology. Moreover, it does not address the same security needs as the TCSEC (e.g., multi-level security is not addressed while on the other hand integrity of information is added). Finally, the interaction with TCSEC-environment(s) is not clear. For these reasons, the TNI is not mandatory for use in the US defense world.

#### 7.3.2 Architecture and services

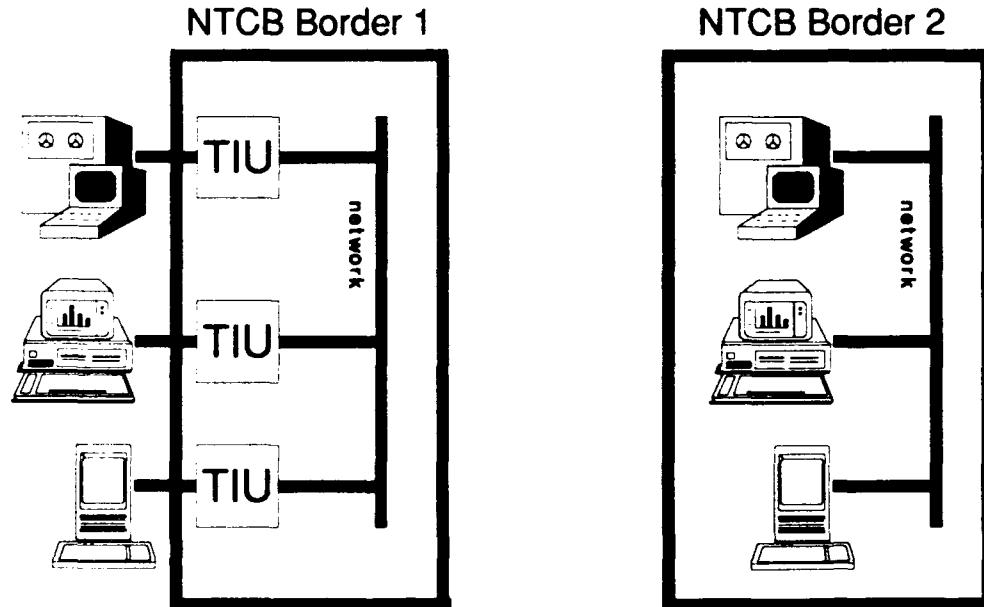
The TNI recognises two views on network security:

- 1 The network is seen as a collection of possibly independently managed end systems (named *hosts*) using a shared communication means. The security at the end systems may differ significantly. The TNI states that 'it might not be practical to evaluate such a network using this Interpretation' [56, page xiii]. Since the TNI does not offer suitable and concrete security criteria with respect to this view, this view is not discussed any further in this section. However, the situation as it is described seems to be quite normal.
- 2 The network is seen as a single unified system that is managed as a whole and implements one single security policy. In this way, a common level of trust is offered throughout the network. A 'single trusted system' network implements a reference monitor which creates a single trusted computing base, named the Network Trusted Computing Base (NTCB), also referred to as Trusted Networking Base (TNB). The NTCB is distributed among the trusted network components.

It is important to note the borders of the NTCB, see figure 13. End systems operating independently of the network and enforcing their own security policy (i.e. have a TCB of their own) lie outside the NTCB (NTCB border 1 in figure 13). Communication between the end systems and the NTCB takes place through a Trusted Interface Unit (TIU) which locally holds a partition of the NTCB and implements a Reference Monitor.

At end systems that are part of the NTCB, the security policy is enforced by the NTCB (NTCB border 2 in figure 13).

In both views, there is no relation between a TCSEC classification of the end systems and the possible TNI classification.



TIU: Trusted Interface Unit

Figure 13: Possible borders of the TNI NTCB

The TNI recognises two fundamental differences between security in the network and in stand-alone situations:

- 1 The communication paths in the network are more vulnerable.

For this reason, the TNI had to counter additional threats that were not in the TCSEC.

Integrity was added as a major issue.

- 2 Individual components in the network are operating concurrently and asynchronously.

Therefore, there is no natural 'single state' in the network. The Reference Monitor needs a single state to be able to enforce security.

This problem is left to a large extent to the manufacturers. In view of the inner border (NTCB border 1 in figure 13) the TNI seems to be in favour of a physically protected TIU that contains enough information of the NTCB to enforce the Network Security Policy. Each TIU implements a Reference Monitor.

The security architecture in the TNI is based on three concepts: a security-enforcing service, defined security policies and the network trusted computing base. These concepts are concisely described below.

#### The security-enforcing service

This service must mediate all activities in the network, decide whether the requested activity is conforming the security policies and enforce this decision. The criteria depend on the TNI class which is the target for the evaluation. This service might be implemented as a (distributed) Reference Monitor. As described above, it is left to the manufacturers how to implement such a mechanism.

#### Security policies

Two policies are addressed. The secrecy policy is a statement about the protection against unauthorised disclosure. An example of a secrecy policy is the Bell - LaPadula policy. The integrity policy is a statement about the prevention of unauthorised modification of information or other manipulations of the communication. An example of an integrity policy is the Clark-Wilson policy [60].

#### Network Trusted Computing Base

The NTCB already is briefly described above as the equivalent of the TCB in a network. The NTCB is defined as the whole of protection mechanisms within a network system, including hardware and software, the combination of which is responsible for enforcing a security policy.

### 7.3.2.1 Security functionality

The TNI addresses two sorts of security functionality: *control objectives* and *optional security services*. Both are briefly described below.

#### *Control objectives*

The control objectives are mandatory and equivalent to those in the TCSEC. These are: security policy, accountability and assurance.

##### Security Policy

The secrecy policy enforces non-disclosure (confidentiality). The integrity policy offers data integrity.

These policies are enforced by discretionary and/or mandatory access control. To support the access control enforcing functions, the TNI describes several supporting mechanisms as the wiping of information (prevention of uncontrolled object reuse) and sensitivity labelling both for confidentiality and integrity of all entities. It remains an open question how an integrity policy can be achieved by means of access control only.

##### Accountability

Accountability in the TCSEC definition is the possibility to keep a person responsible for his actions (blame-ability). In the TNI this depends on the borders of the NTCB. When this border is border 1 of figure 13 the granularity of accountability will be limited to the identity of the end system that is served by a TIU.

To enable accountability, the TCSEC demands identification and authentication of users and an audit on actions that effect classified or sensitive information. In the TNI, accountability of a 'user', being an identified person, might not be achievable. In addition, the TNI demands the auditing of events that effect the integrity policy.

##### Assurance

The third control objective is concerned with guaranteeing or providing confidence that the two security policies are correctly implemented and that the NTCB does indeed accurately mediate and enforce the intent for these policies.

*Optional security services*

The TNI added additional security services to line up as much as possible with the OSI Security Architecture (see section 7.1) which was already available as a Working Draft. Their use is optional. These services are:

**Communications Integrity**

This service is supported by mechanisms for authentication, communications field integrity and non-repudiation.

**Compromise Protection**

This service is supported by mechanisms for data confidentiality, traffic flow confidentiality and selective routing.

**Denial of Service**

This service addresses the protection of availability. It addresses continuity of operations, specific protocol based protection and network management. (Note: this TNI service is not offered by the OSI Security Architecture).

**7.3.3 Openness and relations with other security-providing parts in the system**

The protection of the TNI environment itself depends on physical measures, which are assumed to be present.

When the border of the NTCB is border 2 in figure 13, all information is always protected by the NTCB. The TNI gives insufficient guidance to elaborate further on this approach.

In the case that the border of the NTCB is border 1, the security of the information also depends on the security at the end systems (operating systems and applications). In this case, the end systems' security could be based on the TCSEC. The relations between end system security and network security are not addressed.

Thus, it does not seem extremely promising to use the TNI approach in open systems.

**7.3.4 Which security requirements are addressed**

The following list summarises whether the remaining requirements of section 2.5 can be fulfilled or not, with the functionality as described in the TNI. A distinction is made between borders 1 and 2 of figure 13 where appropriate. It is repeated here that the TNI does not give sufficient guidance for a 'border 2'-approach.

**Representation of real-life tasks and responsibilities in the system**

Border 1: Outside the scope of the NTCB.

Border 2: Same approach as in the TCSEC (hardly addressed).

**(Granularity of) mapping real-life tasks to services**

Border 1: Outside the scope of the NTCB.

Border 2: Same approach as in the TCSEC.

**(Granularity of) mapping responsibilities to rights and duties to services and information**

Border 1: Outside the scope of the NTCB.

Border 2: Same approach as in the TCSEC.

**Authentication information**

The TNI defines an authentication service for users and end systems.

**Security information (rights/duties)**

The security information is distributed over the network and stored in partitioned NTCB components in a distributed security reference database.

**Availability of services and information**

The TNI defines Denial of Service-services.

**Confidentiality of services and information**

The TNI defines confidentiality services and a traffic flow confidentiality service.

**Integrity of services and information**

Integrity services are defined.

**Prevention**

Prevention of loss of confidentiality and integrity. Protection against 'denial of service'.

**Reduction**

One of the integrity services supports recovery for which reductive measures are defined.

Denial of Service-services are able to reserve resources to provide alternate routing.

**Detection**

Audit functions are identical to those defined in the TCSEC.

Furthermore, services are defined that detect integrity and availability breaches.

**Repression**

Not addressed.

**Correction**

One of the integrity services supports recovery.

The denial of service services can activate alternate resources.

### Evaluation

Not addressed.

### Mutual trust between users and system

The NTCB does not have to trust the users since it enforces trust on them. The users have no means to verify the proper operation of the NTCB.

### Distribution of trust in the system

The NTCB is defined as a distributed, trusted system.

### Trust relations between the elements of the system

The NTCB does not have to trust the end systems:

- 1 When the end system is part of the NTCB it is part of the trusted functionality.
- 2 When the end system lies outside the NTCB the NTCB only has a limited responsibility for what happens with the information at the end systems, even though this information is transported via the NTCB (e.g. the secrecy check will be based on the allowed range of labels for that end system and not on that of a specific user).

### Requirements for security imposed by society:

Not a target of the TNI.

A summarising overview of the fulfilment of requirements by the TNI is listed in table 8.

Table 8: Security requirements addressed in the TNI

<u>Requirement</u>	<u>Addressed?</u>
Real-life tasks -> Services	No
Rights/Responsibilities	Border 2 only: hardly
Duties/obligations	No
Exclusions	No
Control of use of services	Border 1: Yes, only network services Border 2: Yes, all services in the system
Control access to information	Yes, granularity: border 1: network unit border 2: not clear from TNI text
Authentication	Yes
Security information	In NTCB distributed databases
Security management	Hardly
Confidentiality	Yes
Integrity	Yes
Availability	Yes
Prevention	Yes
Reduction	Yes
Detection	Yes
Repression	No
Correction	Yes, partly
Evaluation	No
Mutual trust	No
Requirements from society	No target
Focus on	Network
Aware of	-
Horizontal security	Implicit in NTCB
Vertical security	No
Distribution of trust	Yes: definition of NTCB
Trust relations	No

## 7.4 MIT Athena Project: Kerberos

### 7.4.1 Background

Kerberos is an authentication system for unprotected network environments [35, 36]. The authentication server is developed as part of the 'Athena Project' at the US Massachusetts Institute of Technology (MIT). The name Kerberos stems from the three-headed guard dog of Hades, the 'hell hound' in Greek mythology. The first prototype of Kerberos became operational in 1986. Kerberos is gaining more and more acceptance in the USA, especially in the university environment, since the Kerberos protocols are distributed freely (since September/October 1991 an export license is required to export Kerberos outside the USA).

#### 7.4.2 Architecture and services

Kerberos provides a means of verifying the identities of subjects like real users, applications (services) and workstations (hosts) in an untrusted network. Kerberos can be used as the central bookkeeper for security attributes (privileges, rights, etc) and for the distribution of encryption keys as well. It is network orientated, thus it does not assist in the local authentication process. Kerberos performs authentication as a trusted third party by using secret key cryptography.

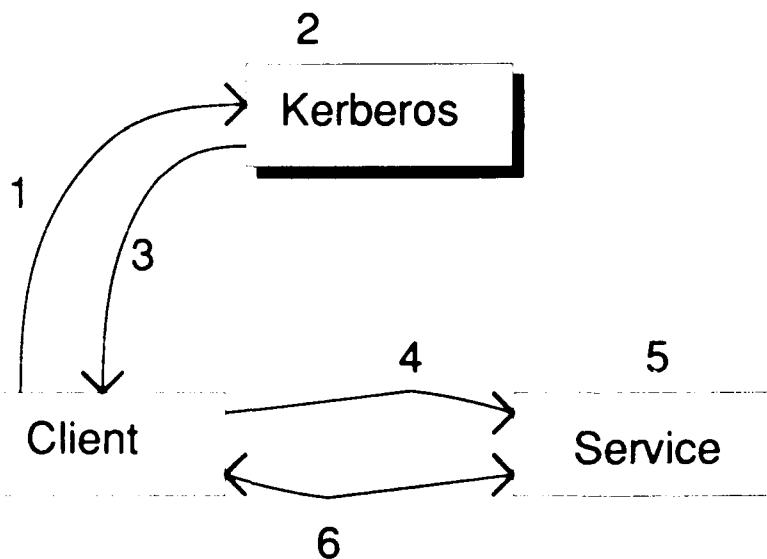


Figure 14: Simplified authentication process of Kerberos

The simplified authentication process is as follows (see figure 14):

- 1 A client (a subject) sends a request to Kerberos (the authentication server) requesting 'credentials' (access rights) for a given network service (the object). This request is encrypted using the client's secret key.
- 2 Kerberos checks the identity based on the client's secret key.

- 3 Kerberos sends a response encrypted with the client's secret key. It contains:
  - The requested credentials in the form of a certificate with a 'ticket' for the given service (the ticket contains the client's identity and a copy of the encryption key for this session, both encrypted with the service's secret key),
  - The encryption key for this session. This *session key* is only valid for this unique session and during a limited time.
- 4 The client sends the ticket to the service he wants to have access to.
- 5 There the ticket is decrypted with the service's secret key. It is verified that it is an uncorrupted message and therefore must be signed by Kerberos.
- 6 Client and server communicate using the session key.

In addition to the checks above, it is also verified that messages are no play-backs of previously recorded communication. Therefore a timestamp is included in the ticket.

#### 7.4.3 Openness and relations with other security-providing parts in the system

The communication means are assumed to be insecure. Kerberos assumes that the Kerberos-server is physically secured and that the Kerberos-software is not corrupted. Furthermore, the security of Kerberos depends on more or less synchronised clocks at the end systems to enable the checking of the timestamps. Finally, for several reasons the security of the end systems remains of importance (although the security of the end systems does not *directly* effect the Kerberos system). The first reason is that the secret keys of applications and operating systems must remain secret, and these are stored in the end systems. Secondly, for practicality's sake, the secret keys of the users will be stored in end systems too. The security of these keys depends on the security at the end systems where the keys are stored.

In [58] some more limitations of the Kerberos system are discussed.

Kerberos aims at horizontal security, but at several vertical levels: user to service, service to service, system to system. Kerberos does not directly address the needs for security in open systems. Nevertheless, Kerberos may be a suitable building block in the provision of distribution of security in open systems.

#### 7.4.4 Which security requirements are addressed

Kerberos only addresses authentication in a network environment. Kerberos can also assist in the management of security attributes and the distribution of encryption keys. Moreover, based on Kerberos many additional security services may be offered.

## 8

## INITIATIVES THAT ADDRESS SECURITY IN SYSTEMS AS A WHOLE

## 8.1 Information Technology Security Evaluation Criteria (ITSEC)

## 8.1.1 Background

The Information Technology Security Evaluation Criteria (ITSEC) is developed by four European countries [31]. This effort, which started in 1989, aims at harmonising security evaluation criteria. From mid 1991, a version of the ITSEC is available for a trial period [30]. The development of the ITSEC is supported by the Commission of the European Communities (CEC).

Under development from the mid of 1991 is the evaluators handbook, called Information Technology Security Evaluation Manual (ITSEM) [32, limited release]. Reasons for the development of the ITSEC were the absence of evaluation criteria that address today's security needs, the problems that arose with the acceptance of evaluations performed in other countries and the difficulty for European manufacturers to have their products accepted by the (American) NCSC for evaluation against the TCSEC-criteria (see also [67]).

The ITSEC-initiative has gained strong support so far. It is anticipated that criteria stated by the ITSEC will be used as a set of *design* criteria for secure systems as well.

The ITSEC is evaluating systems as a whole and concentrates on technical security. Example Targets Of Evaluation (TOE) are (combinations of): applications, operating systems and networks. Targets can be evaluated on a 'as-is' basis (in ITSEC-terms: a product), or within their eventual operational environment which usually means a combination of applications, operating system, networks in a specific environment (in ITSEC-terms: a system).

## 8.1.2 Architecture and services

The ITSEC does not dictate a specific security policy or specific security services. Almost all methods to provide security are permissible as long as the claims regarding the security of the TOE can be evaluated. The structure of the ITSEC document is shown in figure 15. For the purpose of this report and in view of the current state of ITSEC, it is sufficient to know which security policies, architectures and services ITSEC is aware of.

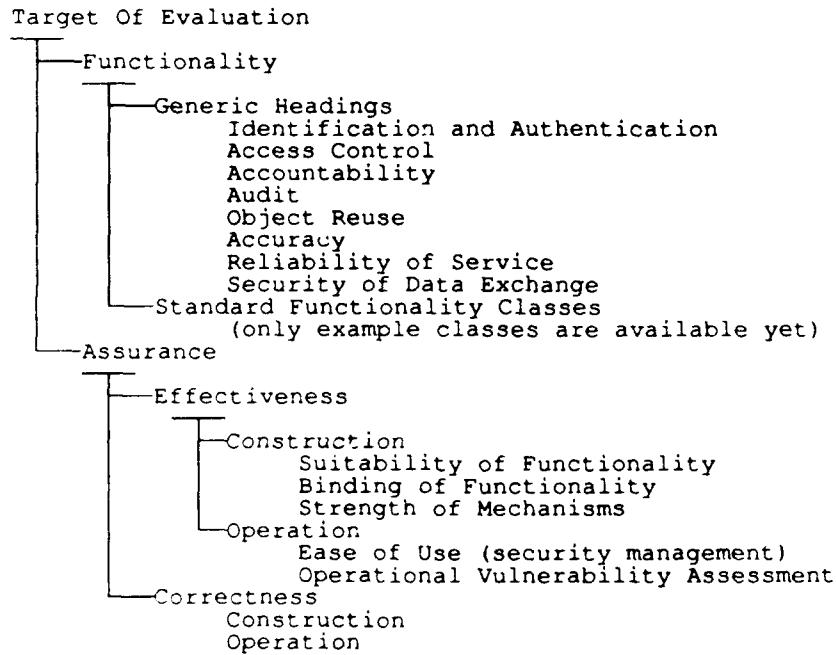


Figure 15: ITSEC Structure

#### 8.1.2.1 Policies

In the higher evaluation levels, a TOE must implement an underlying model of a security policy.

As examples, ITSEC mentions the following security policies:

- 1 The Bell-LaPadula security policy model (see section 6.1).
- 2 The Clark-Wilson security policy model (see section 5.1).
- 3 The Brewer-Nassi security policy model. According to [30] this policy models access control requirements for client confidentiality, typical of a financial services institution.
- 4 The Eizenbergs security policy model. According to [30] this is a policy that models access control rights that vary with time.
- 5 The Landwehr security policy model [63] models data exchange requirements of message handling systems in a network.

Policy 5 is a refinement of policy 1 for a specific environment. Following [30], policies 3 and 4 can be viewed as refinements of policies 1 and 2.

#### 8.1.2.2 Other architectural issues

Again, ITSEC does not dictate a specific approach. The following architectural issues are evaluated:

- Suitability of Functionality: will the security-enforcing functions in fact counter the identified threats?
- Binding of Functionality: how do the security-enforcing functions relate to each other? Do they support one another and do they offer integrated and effective security? One of the methods to achieve Binding of Functionality is to offer a TCSEC TCB.
- Strength of Mechanisms: resistance against direct attacks, e.g. by guessing passwords or by guessing encryption keys of encrypted data.

#### 8.1.2.3 Security Services

ITSEC has two ways to define security functionality. Firstly, a set of security services is described under the 'generic headings'. An evaluation can take place against these security services (extensions of this list should be possible). Secondly, certain clusters of security-enforcing functions for a common environment can be defined in the 'Standard Functionality Classes' (only some example classes are available during the trial period). A manufacturer does not have to define separate claims for all the security services but may claim to conform to a Standard Functionality Class.

The following set of security services is mentioned under the generic headings:

- identification and authentication,
- access control,
- accountability,
- audit,
- object reuse,
- accuracy,
- reliability of service,
- security of data exchange.

Again, a manufacturer has the liberty to define additional security-enforcing functions.

#### 8.1.3 Openness and relations with other security-providing parts in the system

Dependency on other security-providing parts is part of the evaluation, e.g. the vulnerability assessment.

The ITSEC evaluates systems as a whole. The discussion on reuse of previous evaluation results is in progress. Open systems are composed of many elements, some of which may have been evaluated before. Thus, the possibility to reuse evaluation results would make the ITSEC-evaluation of open systems much more practical.

#### 8.1.4 Which security requirements are addressed

The following list primarily is an indication of what security requirements as stated in section 2.5 ITSEC is aware of. As said before, manufacturers can address those security requirements that they think will satisfy the needs of their customers, whether they are mentioned in the ITSEC or not.

##### Representation of real-life tasks and responsibilities in the system

Indirectly addressed for TOEs in a specific operational environment. Not explicitly mentioned, e.g. as a service in ITSEC.

##### (Granularity of) mapping real-life tasks to services

Not explicitly mentioned in ITSEC.

##### (Granularity of) mapping responsibilities to rights and duties to services and information

Not explicitly mentioned in ITSEC.

##### Authentication information

Defined as a service.

##### Security information (rights/duties)

Not explicitly mentioned in the ITSEC. The suitability of the security information in view of the intended use of the TOE is evaluated indirectly as part of the Binding of Functionality.

Security Management is an evaluation issue.

##### Availability of services and information

ITSEC defines a service named 'reliability of services'.

##### Confidentiality of services and information

Confidentiality of information is addressed. Confidentiality of services is not addressed.

##### Integrity of services and information

Several integrity services for information may be captured under the Generic Headings. As far as it concerns the security functions themselves, integrity of services must be achieved in the binding of security functionality.

**Prevention**

Prevention services for the three security aspects confidentiality, integrity and availability may be placed under Generic Headings.

**Reduction**

Not addressed.

**Detection**

Audit is one of the Generic Headings.

**Repression**

Not addressed.

**Correction**

Not addressed.

**Evaluation**

Not addressed.

**Mutual trust between users and system**

Not addressed.

**Distribution of trust in the system**

ITSEC evaluates systems as a whole, including composite systems. Work is in progress on the reuse of evaluation results.

**Trust relations between the elements of the system**

The examination of trust relations is part of the evaluation process.

**Requirements for security imposed by society**

Not identified as potential requirements.

An overview of the requirements that are explicitly mentioned in the ITSEC and thus are recognised as possible requirements is listed in table 9.

Table 9: Security requirements that ITSEC is aware of

<u>Requirement</u>	<u>Is ITSEC aware of this requirement?</u>
Real-life tasks -> Services	Yes, indirectly for TOEs evaluated in their operational environment
Rights/Responsibilities	No
Duties/obligations	No
Exclusions	No
Control of use of services	No
Control access to information	Yes, granularity depends on the product
Authentication	Yes
Security information	No
Security management	Evaluation issue
Confidentiality	Yes
Integrity	Yes
Availability	Yes
Prevention	Yes
Reduction	No
Detection	Partly through audit
Repression	No
Correction	No
Evaluation	No
Mutual trust	No
Requirements from society	No
Focus on	Whole systems
Aware of	-
Horizontal/vertical security	No
Distribution of trust	No, only evaluation of systems as a whole
Trust relations	Part of the evaluation process

## 9

## DISCUSSION AND CONCLUSIONS

This section summarises the discussions in the previous sections and contains the major conclusions. Firstly, the possibility to fulfil the security requirements with one of the initiatives is discussed. Secondly, the relations between the initiatives are discussed as well as the possibility for a coexistence of different approaches in one system. Finally, it is discussed which combinations may be mutually beneficial and may offer a good basis for secure open systems.

#### 9.1 Fulfilment of security requirements

##### Security requirements regarding openness:

For their security, the elements of an open system depend on each other to a large extent. None of the initiatives provides a solid basis for the distribution of security functionality between the elements of an open system (networks, operating systems, databases and other applications). Also missing is the possibility to exchange security information between the elements of the open system. Horizontal security is only found in networks and in some applications (application specific). Vertical security is only addressed in the TDI. The TDI does not define the interfaces that are needed to compose secure systems of 'open' elements. From this, it follows that an element of an open system has no knowledge of the provision of security in its environment. This is a serious problem since most of the elements of an open system are unable to provide the required security by themselves.

However, some of the initiatives (most notable: the ECMA Framework) suggest the concept of *security domains* (a technically bounded group of manageable entities to which a single security policy applies) on a per-element basis. This concept may be a suitable basis to offer services for the exchange of security information and distribution of trust between the elements of the open system. Moreover, the TDI introduces the concept of TCB subsets. TCB subsets enable the local provision of security in the operating system or in an application and offer a promising mechanism for vertical security (if properly defined TCB subset interfaces were available).

##### Requirements for information security that stem from organisational considerations:

None of the initiatives addresses relations between the tasks of an employee in real life and his/her work using the system (whether the system is stand-alone or in a network). The same holds for the mapping of real-life responsibilities to the system.

All the initiatives completely disregard the security functionality that is needed to map normal organisational structures and responsibilities.

Demands from society for information security:

None of the initiatives acknowledges security requirements that stem from relations with society. Services for privacy, legal or other proof of correct functionality and anonymity are not considered (an exception is DAF-security which identifies a need for anonymity, see section 5.2).

Technical security seems to ignore the needs for security that stem from society.

Basic security functionality in the system

Authentication

- Authentication of users is commonly done on a per-computer-system basis (limited to one operating system at one end system). Authentication that can be used over more computer-systems in a network only is addressed in the approaches of the ECMA Framework and Kerberos. Authentication that can be used both in operating systems and applications is addressed in the TDI.
- Many of the initiatives disregard the need for authentication of active entities other than users (applications, services, processes). The need for authentication of passive entities (entities that are being accessed) is disregarded as well. The ECMA Framework is an exception to this.

Management

None of the initiatives offers a structured approach to the management of security information. Only OSI provides a mechanism for the management of security information in its Management Framework.

The properties of information

- Confidentiality is present in all initiatives.
- Integrity is addressed in most of the initiatives. Some of the initiatives suggest an approach that may be applicable to others as well (e.g. basing access-control-decisions on the triplet USER/APPLICATION/INFORMATION).
- Availability is not regarded as an important issue (except by the TNI).

Security measures

- Prevention is the starting point for security in all initiatives.
- Reduction is scarcely addressed and if at all, in an unstructured way.
- In most initiatives detection is synonymous with audit which is always *post factum* and often belated.

- Repression is hardly addressed and if addressed at all, it is incomplete and in an unstructured way.
- Correction is addressed by some of the initiatives, but not in a structured way and with insufficient detail.
- Evaluation is not addressed at all.

From the above it can be concluded that emphasis is put on prevention and that other security measures are neglected to a large extent, and, if addressed, they lack structure.

#### Mutual trust

The users of the system have no means to assure themselves of the proper behaviour of the system. In most cases, the system does not have to trust its users. Some of the initiatives assume that this unbalanced situation is corrected by physical and organisational security measures. In most of the initiatives this problem is disregarded.

### 9.2 How do the different initiatives fit together?

In most systems application security, operating systems security and network security are all needed together. In this section, it is discussed which initiatives do and do not fit properly in one system. In some cases, the initiatives take conflicting approaches, in others, a combination of approaches may be of mutual benefit.

First, it is noted that the ITSEC is not listed below. The ITSEC is a general framework for the specification of security provisions in a TOE. Therefore, it does not show a specific relationship with any of the initiatives, nor does it conflict with any of them.

#### 9.2.1 Applications and Operating System or Networks

Some standards for applications offer hooks to add security functionality at a later stage. The placement of these hooks turns out to restrict the security services that can be offered.

Security that is offered in the OSI applications may easily conflict with the TCSEC approach. Firstly, the security functions that are offered via the hooks of the (non-trusted) applications lie partly outside the Trusted Computing Base (TCB, the bounded group of all security-enforcing functions). The TCSEC approach demands that all security-enforcing functionality be placed inside the TCB. Secondly, and even more important, OSI applications may cause an uncontrolled information flow and frustrate the TCSEC Formal Security Policy (or relaxed versions thereof that can be found in other initiatives).

All the applications that were studied are OSI orientated. However, there is no relation between the security offered by these applications and the security that is available through the OSI Security Architecture (fortunately, there is no conflict either).

The approach in the ECMA Framework conflicts in many aspects with the TCSEC approach. To name the most important ones: firstly, the TCSEC needs a Trusted Computing Base (a trusted 'kernel') whereas the ECMA approach chooses distributed security based on cryptographically protected credentials. Secondly, the ECMA Facilities do not offer the TCSEC 'control objectives'. Thirdly, the ECMA approach offers freedom of choice of a security policy whereas the TCSEC imposes the Bell - LaPadula policy.

The Distributed Application Framework takes an approach that is based on both the ECMA Framework and the OSI SA.

The TDI offers an integrated approach to security in applications and the operating system. Some of the concepts of the TCSEC, on which the TDI is built, had to be extended. It is not easy to extend the TDI approach to networks. The TDI does not match the OSI SA.

The POSIX Security Interface matches the TCSEC in many aspects. Nevertheless, the POSIX Security Interface seems to collide with the TCSEC when the enforcement of privileges is not mediated by the TCB. The POSIX Security Interface offers less security services than the TCSEC and does not offer all of the control objectives of the TCSEC. Network security is not addressed at all by the POSIX Security Interface.

X.509 (The Directory, authentication) may offer a suitable vehicle for authentication services in both the OSI SA and the ECMA Framework approaches. Also, Kerberos can use the same approach as X.509. However, Kerberos is protocol dependent (TCP/IP) and currently cannot be used in a network that uses OSI protocols at the lower layers.

None of the initiatives shows a clear relation to the TNI approach.

#### 9.2.2 Operating System and Network

The TCSEC addresses end systems and not networks. OSI addresses networks and not end systems. It is clear that these two must be fitted together, given the fact that they will have to

coexist in one system since both operating system security and network security are needed. From this report it is concluded that there may be a possibility in the combination of the reference monitor databases and the local part of the SMIB.

The TNI approach aims at offering network security but it does not offer a clear connection with the end systems security (which is the TCSEC-environment).

### 9.3            Mutually beneficial approaches as a starting point for secure open systems

None of the initiatives addresses all security requirements for secure open systems. Some of them provide a good starting point, especially those that do not exclude additional security services. Some initiatives fit better together than others.

Based on the discussions in the previous sections, it can be concluded that currently only few combinations of initiatives remain that can be used as a starting point to create a firm basis for secure open systems.

A promising combination is:

The ECMA Framework (and DAF) with the OSI SA. This combination may benefit from the use X.500 and/or an approach like that of Kerberos. In this combination, integration with application security and operating system security remains a problem that must be solved.

Another possible combination is:

The combination of TCSEC, POSIX Security Interface and TDI. The POSIX Security Interface should be adapted in such a way that it provides proper TCB subset interfaces that enable the integration of application and operating system security. In this combination the integration with network security is a problem. If network security is based on the TNI, integration with application and operating system security will not be possible.

The latter combination excludes integration of network security and offers far less security functionality than the first. Therefore, it seems reasonable to start further study based on the first combination. However, the potential of the TDI TCB subsets must not be overlooked. The concept of TCB subsets seems may be applied to provide security in the ECMA security domains.

It is concluded that there is a lack of integration between application security, operating system security and network security. Therefore, an architecture for security functionality and interfaces is needed that crosses the borders of applications, operating systems and networks.

The most promising starting point for a basis for secure open systems is the combination of the standards of ECMA and OSI SA, including the use of supporting standards (like X.500 and Kerberos). The problem of integrating application and operating system security must be solved (the TDI TCB subset may provide a suitable mechanism). For the purpose of integration of security, an architecture for security functionality and interfaces is needed that crosses the borders of applications, operating systems and networks.

It must be investigated which of the identified security requirements can be added (and how). Also, if the TCSEC approach appears to be insufficient for operating system security within open systems, a derivative or alternative approach must be developed.

## 10                   ACRONYMS

The following acronyms are used within this document.

ANSI	American National Standards Institute
ASE	Application Service Element
BC	Behavioural Component
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CEC	Commission of the European Communities
DAF	Support Framework for Distributed Applications
DBMS	Database management system
DoD	USA Department of Defense
EC	European Commission
ECMA	European Computer Manufacturers Association
EDI	Electronic Data Interchange
ETSI	European Telecommunications Standards Institute
EWOS	European Workshop for Open Systems
FIPS PUB	Federal Information Processing Standard Publication
FTAM	File Transfer, Access and Management
IEEE	Institute of Electrical and Electronics Engineers
IEPG	Independent European Programme Group
IPSE	Integrated Project Support Environment
ITAEGV	Information Technology Advisory Expert Group for Information Security
ISO	International Organisation for Standardisation
IT	Information Technology
ITSEC	Information Technology Security Evaluation Criteria
ITSEM	Information Technology Security Evaluation Manual
JTC	Joint Technical Committee
LAN	Local Area Network
MHS	Message Handling System
MIT	Massachusetts Institute of Technology
NATO	North Atlantic Treaty Organisation
NCSC	USA National Computer Security Center

<b>NIST</b>	<b>USA National Institute of Standards and Technology</b>
<b>NOSA</b>	<b>NATO OSI Security Architecture</b>
<b>NTCB</b>	<b>Network Trusted Computing Base</b>
<b>ODA</b>	<b>Office Document Architecture</b>
<b>OIW</b>	<b>Open Implementors Workshop</b>
<b>OSF</b>	<b>Open Software Foundation</b>
<b>OSI</b>	<b>Open Systems Interconnection</b>
<b>OSI SA</b>	<b>OSI Security Architecture</b>
<b>PAC</b>	<b>Privilege Attribute Certificate</b>
<b>PCTE+</b>	<b>Portable Common Tool Environment</b>
<b>POSIX</b>	<b>Portable Operating System Interface for Computer Environments</b>
<b>SC</b>	<b>Security Component</b>
<b>SC</b>	<b>Sub Committee</b>
<b>SILS</b>	<b>Standard for Interoperable Local Area Network Security</b>
<b>SMIB</b>	<b>Security Management Information Base</b>
<b>SSA</b>	<b>Supportive Security Application</b>
<b>STANAG</b>	<b>NATO Standard Agreement</b>
<b>TC</b>	<b>Technical Committee</b>
<b>TCB</b>	<b>Trusted Computing Base</b>
<b>TCSEC</b>	<b>Trusted Computer System Evaluation Criteria (the Orange Book)</b>
<b>TDI</b>	<b>Trusted Database Management System Interpretation of the TCSEC (the Grey Book)</b>
<b>TIU</b>	<b>Trusted Interface Unit</b>
<b>TNB</b>	<b>Trusted Networking Base</b>
<b>TNI</b>	<b>Trusted Network Interpretation of the TCSEC (the Red Book)</b>
<b>TOE</b>	<b>Target Of Evaluation</b>
<b>TOS</b>	<b>Text and Office Systems</b>
<b>UIN</b>	<b>User-Identification Number</b>

11            **REFERENCES**

- 1    "Banking - Approved algorithms for message authentication", part 1 and 2, ISO 8731-1/2 : 1987
- 2    "Banking - Key Management (wholesale)", ISO 8732:1988
- 3    "Beveiliging bij Datacommunicatie", NGI, ISBN 90267 1357 6, NUGI 852 (in Dutch), 1989
- 4    "DAF: Security", CCITT Support Framework for Distributed Applications (DAF), DAF: Working Document on Security, version 4, May 1989
- 5    "Department of Defense Trusted Computer System Evaluation Criteria", CSC-STD-001-83 (also known as TCSEC or Orange Book), August 1983, replaced by a revised edition [6] in December 1985
- 6    "Department of Defense Trusted Computer System Evaluation Criteria", DoD 5200.28 STD. (also known as TCSEC or Orange Book), December 1985
- 7    "Electronic data interchange for administration, commerce and transport (EDIFACT)", ISO 9735:1988
- 8    "Framework for the Support of Distributed Application (DAF)", ISO/IEC JTC1 N544 (source CCITT), 1989-09-26
- 9    "Glossary of Information Technology Security Definitions", ISO/IEC JTC1 SC27 N270 (Standing Document), 1991-05-29
- 10   "Guide to Open Systems Security (Revised Draft)", ISO/IEC JTC1/SC21 N6167, August 1991
- 11   "Guide to POSIX Open Systems Environments", IEEE P1003.0/D8, June 1990
- 12   "Information Processing Systems - Open Systems Interconnection - Basic Reference Model", ISO 7498:1984
- 13   "Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture", ISO 7498-2:1988
- 14   "Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management Framework", ISO 7498-4:1989
- 15   "Information Processing Systems - Open Systems Interconnection - File transfer access and management", ISO 8571 parts 1 to 4 (1-10-1988)
- 16   "Information Processing Systems - Open Systems Interconnection - The Directory - Part 8: Authentication Framework", ISO/IEC 9594-8:1990

- 17 "Information Processing, Text and Office Systems, Office Document Architecture (ODA) and interchange format", ISO 8613:1990
- 18 "Information Processing, Text and Office Systems, Office Document Architecture (ODA) and interchange format - Addendum 4: Security", ISO 8613/DAD4, 1990
- 19 "Information Technology - Open Systems Interconnection - Security Frameworks in Open Systems - Part 1: Working Draft Security Frameworks Overview", ISO/IEC JTC1/SC21 WG1/5044, 30 May 1990
- 20 "Information Technology - Open Systems Interconnection - Security Frameworks for Open Systems - Part 2: Authentication Framework", ISO/IEC DIS 10181-2 (1991-07-18)
- 21 "Information Technology - Open Systems Interconnection - Security Frameworks in Open Systems - Part 3: Working Draft Access Control Framework", ISO/IEC JTC1/SC21 N 5045, July 1990
- 22 "Information Technology - Open Systems Interconnection - Security Frameworks in Open Systems - Part 4: - Working Draft Non Repudiation Framework", ISO/IEC JTC1/SC21 N5046, July 1990
- 23 "Information Technology - Open Systems Interconnection - Security Frameworks in Open Systems - Part 5: Working Draft Confidentiality Framework", ISO/IEC JTC1/SC21 N5048, July 1990
- 24 "Information Technology - Open Systems Interconnection - Security Frameworks in Open Systems - Part 6: Working Draft Integrity Framework", ISO/IEC JTC1/SC21 N5047 (July 1990)
- 25 "Information Technology - Open Systems Interconnection - The Directory - Proposed Draft Addendum to ISO 9594 (parts 2 to 4) on Access Control", ISO/IEC 9594-1 to 3 /PDAD 1, also available as ISE/IEC JTC1/SC21 N4041, N4042, N4043, December 1989
- 26 "Information Technology - Telecommunications and exchange between systems - Network layer security protocol", (NLSP), CD 11577:1991, 1991-11-13
- 27 "Information Technology - Telecommunications and exchange between systems - Open Systems Interconnection - Transport layer security protocol Amendment 1: Security association establishment protocol", DIS 10736 PDAM1 december 1991, JTC1/SC6 N6891 Attachment 4, 1991-12-23
- 28 "Information Technology - Telecommunications and information exchange between systems - Lower Layer Security Guidelines, 2nd Working Draft", ISO/IEC JTC1/SC6 N6957, 1991-08-09

- 29 "Information Technology - Telecommunications and information exchange between systems - Transport layer security protocol", (TLSP), DIS 10736:1991 (October), 1991-10-16
- 30 "Information Technology Security Evaluation Criteria", ITSEC version 1.2 (provisional). ISBN 92-826-3004-8, June 1991
- 31 "Information Technology Security Evaluation Criteria", ITSEC version 1.1, January 1991
- 32 "Information Technology Security Evaluation Manual (ITSEM)", draft version 0.1, December 1991 (limited distribution)
- 33 "Introducing PCTE+", Independent European Program Group Technical Area 13, April 1989
- 34 "ISO/TC68 Security Standards Work Program", ISO/TC68 N396, 22 June 1990
- 35 "Kerberos version 5 RFC", December 1990, available from krb-protocol@athena.mit.edu
- 36 "Kerberos: An authentication Service for Open Network Systems", MIT Project Athena, Cambridge MA 02139, January 1988
- 37 "Message Handling Systems", CCITT Recommendation X.400-X.430, 1988
- 38 "Message Oriented Text Interchange Systems (MOTIS)", ISO/IEC DIS 10021, May 1988
- 39 "NATO OSI Security Architecture (NOSA)", NATO AC/302 TSGCEE (SG9) Doc-48, 1988
- 40 "NATO Reference Model for Open System Interconnection", STANAG 4250:1986
- 41 "Portable Common Tool Environment (PCTE) - Abstract Specification", ECMA Standard 149, December 1990
- 42 "POSIX", IEEE P1003.1a/D5, June 1990 (also available as DIS 9945)
- 43 "Security Application - Authentication and Privilege Attribute", ECMA/TC32-TG9/91/26 (7th draft, April 1991)
- 44 "Security Architecture for NATO Information Systems Interconnection (SANISI)", NATO AC/302 TSGCEE (SG9) Doc-53, 1989 (NATO CONFIDENTIAL)
- 45 "Security in a Distributed Computing Environment", Open Software Foundation (OSF) White Paper, January 1991
- 46 "Security in Open Systems - A Security Framework", ECMA TR/46, July 1988
- 47 "Security in Open Systems - Data Elements and Service Definitions", ECMA-138, December 1989
- 48 "Security in the OSF/1 Operating System", Open Software Foundation (OSF) White Paper, November 1990
- 49 "Security Interface for POSIX", IEEE P1003.6/D7, August 1990
- 50 "Standard for Interoperable Local Area Network (LAN) Security (SILS)", IEEE 802.10B/D6, ISO/IEC JTC1/SC6 N6596, 1990-11-16
- 51 "Summary of projects in the SC27 Programme of Work", ISO/IEC JTC1/SC27 N244, 1991

- 52 "Taxonomy for Security Standardisation"; September 90; CEN/CENELEC Security Group; CSecG/49/90; (also available as SC27 N173 and JTC1 N1040)
- 53 "Taxonomy of Security Standardisation"; Version 1.1, November 1991; ITAEGV N23; Ted Humphreys (Ed.) (also available as EWOS/EGSEC/91/027). Limited distribution
- 54 "Third Working Draft Upper Layers Security Model", ISO/IEC JTC 1/SC21 N5001 (1990-06-05)
- 55 "Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria", (also known as TDI or Grey Book), USA National Computer Security Center, April 1991, NCSC-TG-021, library no. S235.625
- 56 "Trusted Network Interpretation of the trusted computer system evaluation criteria", (also known as TNI or Red Book), USA National Computer Security Center, July 1987, NCSC-TG-005, lib.nr. S228,526 Version 1
- 57 Bell D.E, LaPadula L.J, "Secure Computer Systems: Unified Exposition and Multics Interpretation", Report MTR-2997 Rev.1, MITRE, 1976
- 58 Bellovin S.M, Merrit M, "Limitations of the Kerberos Authentication System", Conference Proceedings, USENIX - Winter '91 - Dallas TX, 1991
- 59 Boonstra A, "Politieke aspecten bij de ontwikkeling van informatiesystemen", INFORMATIE, Vol 33, No 12, December 1991 (in Dutch)
- 60 Clark D.D, Wilson D.R, "A Comparison of Commercial and Military Computer Security Policies", Proceedings if the IEEE Symposium on Security and Privacy, Oakland, April 1987
- 61 I'Anson C, Mitchell C, "Security Defects in CCITT Recommendation X.509", Computer Communication Review, Vol 20, nr 2, p30-34
- 62 Karila Arto T, "Open Systems Security - an Architectural Framework", Telecom Finland, Helsinki, the Finnish Government Printing Centre, ISBN 952-90-2783-4, 1991
- 63 Landwehr C.E, McLean J, "A security model for military message systems"; ACM Transactions on Computer Systems, Vol 2, nr 3, p198-222, August 1984
- 64 Luijif H, Overbeek P. "The FEL-TNO uniform open systems model", pp. 201-208. In: Towards an Open World: Proceedings 1989 DECUS Europe Symposium, The Hague, Holland, September 18-22, 1989. DECUS, Switzerland.
- 65 Overbeek P, Luijif H, "UNIFORM OPEN SYSTEMS model: A network wide view on applications and operating systems", pp. 1/112 - 1/134. In ECODU, (Ed.), Conference Proceedings ECODU 47, European Control Data Users, Z.pl., April 17-21 1989.

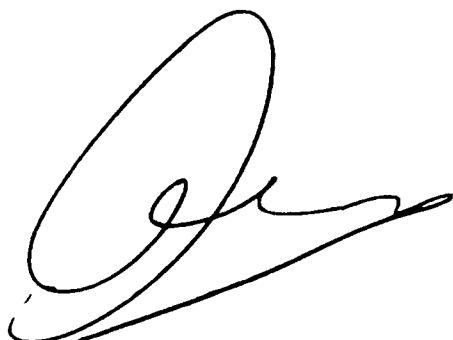
- 66 Overbeek P.L, "Information Security: Past, Present and Future", Securicom '91; also available as TNO Report FEL-91-B100, 1991
- 67 Overbeek P.L, "OSI Security and Relations with other Security Standards", in: Shape Technical Centre, Proceedings OSI Symposium 1990, SP-8 volume 2; also available as TNO Report FEL-91-B99, 1991
- 68 Overbeek P.L, "Review Draft POSIX Security Interface, P1003.6/D7", TNO FEL U/9104236, also available as NNI Doc.Nr. 22/91-32 and NNI Dcc.Nr 27/91-25, 1991
- 69 Parker T.A, "Application Access Control Standards", Computers & Security vol 9, nr 6, ISSN 0167-4048, October 1990

---



D.W. Fikkert  
(project manager)

---

  
ir. H.A.M. Luijif  
(supervisor)

---

ir. P.L. Overbeek  
(author)

**Appendix**Page  
A.1**APPENDIX A: INDEX**

This index lists some of the key words used in this report with a reference to the pages where they can be found.

**A**

Access control 29, 31-32, 40, 44, 54, 58, 61-62, 64, 66-68, 77, 85-86  
 Accountability 13, 15, 32, 54, 61-62, 65, 77, 85-86  
 Accuracy 85-86  
 Anonymity 13, 15, 39-40, 91  
 ANSI 23, 96  
 ASE 34-35, 96  
 Assurance 9, 32, 54, 77, 85  
 Audit 11, 13, 32, 36, 38, 40, 43, 46, 54, 56, 58, 61-63, 65-66, 68, 71, 73, 77, 79, 85-86, 88-89, 91  
 Authentication 14, 16, 22, 27, 31-34, 36, 38-40, 43, 45, 47-48, 54-58, 61-62, 64-68, 70-71, 73, 77-79, 81-83, 85-87, 89, 91, 93  
 Availability 8, 11, 15-16, 28, 35-36, 38, 45, 56, 58, 64, 66, 70-71, 73, 78-79, 81, 87-89, 91

**B**

Basic Security Theorem 52  
 BC 39, 96  
 Bell - La Padula (security policy) 35, 52-54, 56, 61, 64, 76, 85, 93  
 Billability 13  
 Binding of Functionality 85-87

**Brewer - Nash (security policy)**

85

**C**

CCITT 2, 4, 7, 9, 21, 25, 38, 96  
 CEC 2, 4, 84, 96  
 Certificate 31, 32, 33, 40, 83, 97  
 Clark - Wilson (security policy) 29, 76, 85  
 Communications Integrity 78  
 Compromise Protection 78  
 Confidentiality 11, 13, 15-16, 28, 32, 36, 38, 40, 46, 54, 56, 58, 61, 64-67, 70-71, 73, 77-79, 81, 85, 87-89, 91  
 Control objectives 54, 77, 93  
 Correction 13, 15-16, 37-38, 46, 57-58, 65-66, 71, 73, 79, 81, 88-89, 92

**D**

DAF 7, 38-41, 91, 94, 96  
 DBMS 43, 96  
 Delay 39  
 Denial of service 27-28, 39, 78-79  
 Detection 13, 15-16, 36, 38, 46, 56, 58, 65-66, 68, 71, 73, 79, 81, 88-89, 91  
 Disclosure 27, 61-62, 64, 67, 76-77

---

Appendix	Page
	A.2

<b>Distribution of trust</b>	
16, 25, 35, 37-38, 46-47, 57-58, 63, 65-66, 70, 72-73, 80-81, 88-90	<b>Hosts</b>
	74, 82
<b>DoD</b>	<b>I</b>
2, 4, 9, 21, 49, 59, 74, 96	<b>Identification</b>
<b>Domain (security)</b>	14, 43, 54, 61-62, 64, 66, 77, 85-86, 97
29-31, 33-39, 43-44, 50-52, 68, 72	<b>IEEE</b>
	2, 4, 22-23, 58, 96
<b>E</b>	<b>IEPG</b>
<b>EC</b>	24, 96
96	<b>Information Label</b>
<b>ECMA</b>	61
2, 4, 7, 9, 21, 24-25, 27-30, 32-40, 90-91, 93-96	<b>Inheritance</b>
<b>EDI</b>	32
7, 21, 47, 70, 96	<b>Integrity</b>
<b>Eizenbergs security policy</b>	11, 13, 15-16, 28, 32-33, 36, 38, 40, 45-46, 48, 55-56, 58, 61-62, 65-66, 68, 70-74, 76-79, 81, 87-89, 91
85	<b>Interception</b>
<b>End system</b>	39
29, 33-34, 68-69, 72, 77-78, 80, 91	<b>IPSE</b>
<b>ETSI</b>	24, 96
24, 96	<b>ISO</b>
<b>Evaluation (of security measures)</b>	2, 4, 9, 96
13	<b>IT</b>
<b>Event cycle</b>	17, 21, 49, 96
12	<b>ITAEGV</b>
<b>EWOS</b>	24, 96
24, 96	<b>ITSEC</b>
<b>F</b>	7, 22, 26, 84-89, 92, 96
<b>Facility (security)</b>	<b>ITSEM</b>
29-38	84, 96
<b>FIPS PUB</b>	<b>J</b>
96	<b>JTC</b>
<b>Formal security policy</b>	96
49, 52-53, 56, 92	<b>K</b>
<b>FTAM</b>	<b>Kerberos</b>
7, 47, 70, 96	7, 22, 25, 81-83, 91, 93-95
<b>G</b>	<b>Key</b>
<b>Global requirements</b>	14, 33, 40, 48, 68, 82, 83
44	<b>L</b>
<b>Granularity</b>	<b>Label</b>
14-15, 34-36, 38, 42, 45-47, 55-56, 58-59, 62-64, 66, 70, 73, 77, 79, 81, 87, 89	61
<b>H</b>	<b>Landwehr</b>
<b>Horizontal security</b>	85
16, 18, 35, 38, 47, 58, 63, 66, 70, 73, 81, 83, 90	

<b>Legal proof</b>	<b>Open elements</b>
13, 15	10, 15
<b>Local requirements</b>	<b>Optional security services</b>
44	77-78
<b>M</b>	<b>Organisation</b>
<b>Manipulation</b>	2, 6, 9-11, 13-15, 29, 34-35, 96
39	<b>OSF</b>
<b>Masquerading</b>	24, 97
39	<b>OSI</b>
<b>MHS</b>	8, 97
7, 21, 47, 96	<b>OSI SA</b>
<b>MIT</b>	67-68, 70-73, 94-95, 97
7, 22, 81, 96	<b>P</b>
<b>Mutual trust</b>	<b>PAC</b>
16, 37-38, 46, 57-58, 65-66, 71, 73, 80-81, 88-89,	33, 97
92	<b>PCTE+</b>
<b>N</b>	24, 97
<b>NATO</b>	<b>Physical</b>
2, 4, 7, 9, 22, 25, 73-74, 84, 96, 97	12, 18, 25, 33, 39, 40-41, 55, 57, 70, 78, 92
<b>NCSC</b>	<b>Physically</b>
2, 4, 9, 21-22, 41, 44, 49, 59, 74, 84, 96	39, 76, 83
<b>Networked system</b>	<b>Pipes</b>
17, 18	14, 59
<b>No read up</b>	<b>POSIX</b>
52	7, 22, 26, 58-66, 93-94, 97
<b>No write down</b>	<b>Prelay</b>
52	39
<b>NOSA</b>	<b>Prevention</b>
7, 22, 73-74, 97	3, 12, 15, 16, 36, 38, 46, 56, 58, 65-66, 71, 73,
<b>Notarisation</b>	76-77, 79, 81, 88-89, 91-92
40, 68, 72-73	<b>Privacy</b>
<b>NTCB</b>	11, 13, 15, 91
74-81, 97	<b>Privilege</b>
<b>O</b>	27, 31, 33, 36, 38, 40, 59, 60, 61, 62, 64, 65, 66,
<b>Object</b>	82, 93, 97
28-29, 31, 39, 43, 50-55, 59, 61, 64, 69, 77, 82,	<b>Proof of proper functioning</b>
85, 86	13, 15
<b>ODA</b>	<b>Proof of transactions</b>
22-23, 97	13, 72
<b>OIW</b>	<b>Protection</b>
24, 97	23, 32-34, 42, 44-46, 54-55, 59, 63-64, 67-68, 70,
<b>Open element</b>	76, 78-79
10, 15	

<b>R</b>	
Reduction	
13, 15-16, 36, 38, 46, 56, 58, 65-66, 71, 73, 79, 81, 88-89, 91	Security service 62
Reference monitor	Security services 21-23, 25, 29, 32, 35, 39, 41, 47, 61, 63, 67-69, 72, 74, 78, 83-84, 86, 92-94
42, 45, 47, 49-56, 58-59, 74-76, 94	Security-enforcing functions 54-55, 59, 63, 86, 92
Relay	Separation 11, 35, 44, 56
39	Sequencing 39
Replay	SILS 22-23, 97
39	SMIB 68-73, 94, 97
Repression	Society 3, 6, 13, 15-16, 37-38, 46, 57-58, 66, 72-73, 80-81, 88-89, 91
Repudiation	SSA 35, 97
32, 39-40, 68, 72, 78	STANAG 73, 97
Reuse	Star-Property 52
39, 54, 77, 85-88	Strength of Mechanisms 85-86
Rights and duties	Subject 28-31, 42-44, 46-47, 50-53, 55, 59, 60-61, 64, 82
11, 14-15, 32, 36, 45, 56, 64, 70, 79, 87	Suitability of Functionality 85-86
Roles	<b>T</b>
10-11	TC 23, 97
Routing control	TCB 41-47, 49, 54-60, 62-66, 75-76, 86, 92-93, 97
40	TCB subset 41-47, 90, 94-95
<b>S</b>	TCSEC 7, 21-22, 24-25, 27, 41-43, 45-47, 49-52, 54-59, 62, 74-79, 84, 86, 92-95, 97
SC	TDI 7, 21, 26, 41-45, 47, 90-91, 93-95, 97
39-40, 97	TIU 75-77, 97
Security attributes	TNB 74, 97
31, 40, 82-83	
Security event	
12, 15	
Security information	
15-16, 31-32, 36, 38, 45, 47-48, 51-52, 56, 58, 64-66, 68-69, 70-73, 79, 81, 87, 89-91	
Security Interface	
7, 22, 26, 58-60, 62-66, 93, 94	
Security interfaces	
59-60, 62	
Security management	
16, 32, 38, 58, 66, 68-70, 73, 81, 85, 87, 89, 97	
Security policy	
29-30, 33-35, 39-40, 49, 52, 54, 56, 63, 68, 74-77, 84-85, 90, 92-93	
Security requirements	
2, 6-7, 10, 13-15, 24-25, 27, 35, 37-38, 41, 45, 47, 49, 55, 58, 64, 66, 70, 73, 78, 81, 83, 87, 89-91, 94-95	

**TNI**

7, 22, 25, 74-81, 91, 93-94, 97

**TOE**

84-85, 87, 92, 97

**TOS**

23, 97

**Traffic analysis**

39

**Traffic padding**

40

**Trust relations**

16, 37-38, 43-44, 46-47, 55, 57-58, 65-66, 72-73,  
80-81, 88-89

**Trusted functionality**

29, 33, 54, 68, 80

**Trusted path**

43, 57

**U**

**UIN**

64, 97

**V**

**Vertical security**

15-16, 18-19, 25, 35, 38, 44, 47, 58, 63, 66, 70,  
73, 81, 89-90

UNCLASSIFIED

REPORT DOCUMENTATION PAGE

(MOD-NL)

1 DEFENSE REPORT NUMBER (MOD-NL)	2. RECIPIENT'S ACCESSION NUMBER	3. PERFORMING ORGANIZATION REPORT NUMBER
TD91-3274		FEL-91-B293
4. PROJECT/TASK/WORK UNIT NO.	5. CONTRACT NUMBER	6. REPORT DATE
20555		DECEMBER 1991
7. NUMBER OF PAGES	8. NUMBER OF REFERENCES	9. TYPE OF REPORT AND DATES COVERED
108 (INCL 1 APPENDIX, EXCL RDP & DSTR LIST)	69	FINAL
10. TITLE AND SUBTITLE SECURE OPEN SYSTEMS AN INVESTIGATION		
11 AUTHOR(S) OVERBEEK P.L.		
12 PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) TNO PHYSICS AND ELECTRONICS LABORATORY, P.O. BOX 96864, 2509 JG THE HAGUE, THE NETHERLANDS OUDE WAALSDORPERWEG 63, 2597 AK THE HAGUE, THE NETHERLANDS		
13 SPONSORING/MONITORING AGENCY NAME(S)		
14 SUPPLEMENTARY NOTES		
15 ABSTRACT (MAXIMUM 200 WORDS, 1044 POSITIONS) THIS REPORT OUTLINES THE ACHIEVEMENTS OF CURRENT STANDARDISATION EFFORTS IN THE AREA OF SECURE OPEN SYSTEMS. SECURITY IN OPEN SYSTEMS IS A SPECIAL PROBLEM SINCE ALL ELEMENTS IN AN OPEN SYSTEM (HARDWARE, NETWORKS, OPERATING SYSTEMS, DATABASES AND OTHER APPLICATIONS) MUST BE ABLE TO OFFER THE REQUIRED SECURITY IN CO-ORDINATION WITH EACH OTHER. A NEW VIEW ON SECURITY REQUIREMENTS IS PRESENTED. THE INITIATIVES OF, AMONG OTHERS, CCITT, DOD/NCSC, EC, ECMA, IEEE, ISO AND NATO ARE STUDIED. THE MAIN CONCLUSIONS ARE: 1/ NO INITIATIVE ADDRESSES ALL REQUIREMENTS FOR SECURE OPEN SYSTEMS; 2/ NO INITIATIVE GIVES A SOLID BASIS FOR CO-ORDINATION OF SECURITY AMONG ALL ELEMENTS OF AN OPEN SYSTEM; 3/ SECURITY FUNCTIONALITY THAT IS NEEDED TO MAP NORMAL ORGANISATIONAL STRUCTURES AND RESPONSIBILITIES IS DISREGARDED; 4/ TECHNICAL SECURITY IGNORES THE NEEDS FOR SECURITY THAT STEM FROM SOCIETY; 5/ THE BASIC SECURITY FUNCTIONALITY DESCRIBED IN THE INITIATIVES IS DIVERGENT AND SOMETIMES CONFLICTING; 6/ EMPHASIS IS PUT ON PREVENTION, OTHER POSSIBLE SECURITY MEASURES ARE NEGLECTED; 7/ THERE IS A LACK OF INTEGRATION BETWEEN SECURITY IN APPLICATIONS, THE OPERATING SYSTEM AND THE NETWORK. AN ARCHITECTURE FOR SECURITY FUNCTIONALITY IS NEEDED THAT CROSSES THE BORDERS OF THESE ELEMENTS OF AN OPEN SYSTEM.		
16 DESCRIPTORS INFORMATION SYSTEMS SECURITY DISTRIBUTED NETWORKS STANDARDISATION	IDENTIFIERS	
17a SECURITY CLASSIFICATION (OF REPORT) UNCLASSIFIED	17b. SECURITY CLASSIFICATION (OF PAGE) UNCLASSIFIED	17c. SECURITY CLASSIFICATION (OF ABSTRACT) UNCLASSIFIED
18 DISTRIBUTION/AVAILABILITY STATEMENT UNLIMITED	17d. SECURITY CLASSIFICATION (OF TITLES) UNCLASSIFIED	

UNCLASSIFIED